ForcingDAS: Unified and Robust Data Assimilation via Diffusion Forcing

Yixuan Jia[1] **Siyi Chen**[1] **Yida Pan**[1] **Xiao Li**[1] **Lianghe Shi**[1] **Chanyong Jung**[1] **Haijie Yuan**[2]
Ismail Alkhouri[3,1] **Yue Cynthia Wu**[1] **Saiprasad Ravishankar**[2] **Jeffrey A. Fessler**[1] **Qing Qu**[1]

[1]University of Michigan, Ann Arbor, MI, USA [2]Michigan State University, East Lansing, MI, USA
[3]Los Alamos National Laboratory, Los Alamos, NM, USA

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 are fragile when observations are non-Markovian (when they form only a partial slice of a higher-dimensional latent state as in real-world weather data): they tend to accumulate errors over long horizons. 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, with the largest gains observed on real-world weather benchmarks.

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., 2009; 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) (Taira et al., 2020; 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 $\boldsymbol{\eta}_k \sim \mathcal{N}(\mathbf{0}, \gamma^2 \boldsymbol{I}_M)$.

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).

Correspondence to: Yixuan Jia <jiayx@umich.edu>.
Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

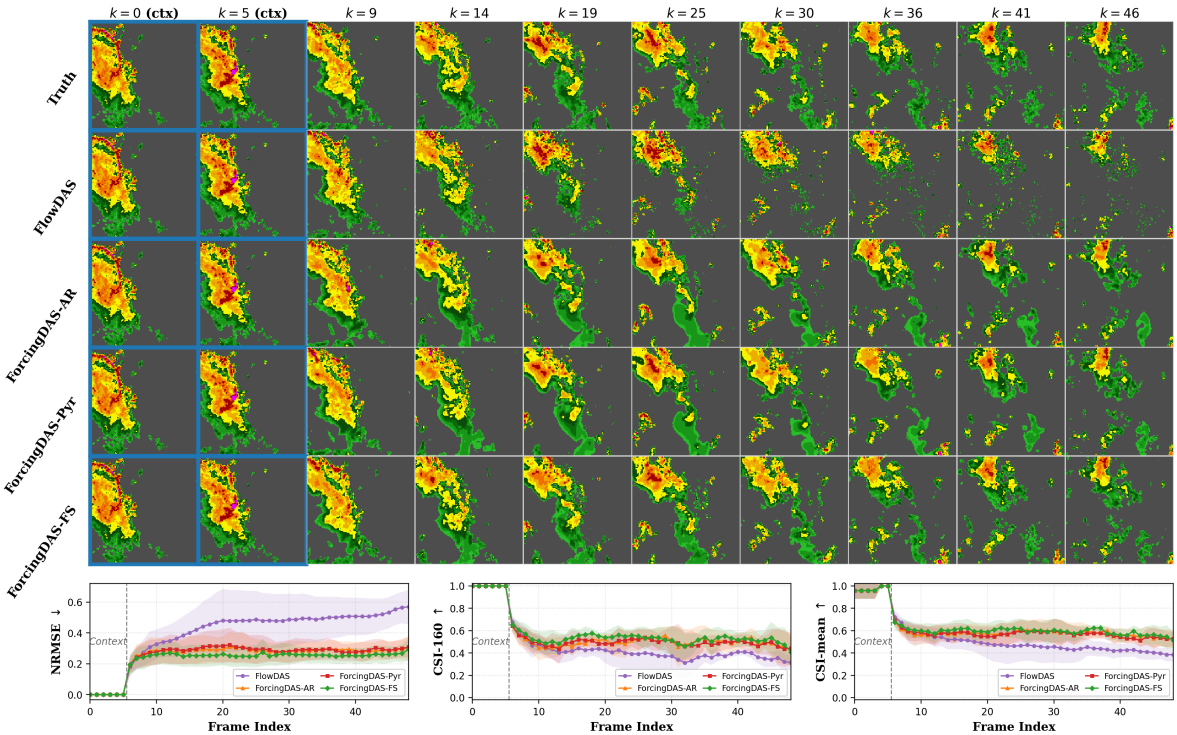

Figure 1. **Demonstration of ForcingDAS on precipitation nowcasting** on a held-out trajectory from the Storm Event Imagery (SEVIR) dataset, Vertically Integrated Liquid (VIL) radar product, under sparse pixel observations (10% of pixels visible) with 6 clean context frames seeding the sequence (blue-bordered columns). **Top:** ground truth and predictions from the per-step learned filter FlowDAS and three inference regimes of a single ForcingDAS model: filtering (AR), fixed-lag smoothing (Pyr), and full-sequence smoothing (FS). **Bottom:** per-frame NRMSE (lower is better), CSI-160 (higher is better), and CSI-mean (higher is better); the vertical line marks the end of the context window. ForcingDAS is *unified*, since the same trained model serves all three regimes without retraining, and *robust over long horizons*: FlowDAS drifts as per-step error accumulates beyond the context, while all ForcingDAS variants track the ground truth across the full trajectory.

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 (Kálmán, 1960; Evensen, 2003) assumes linear-Gaussian dynamics, particle filters (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., 2024c) 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) assume that the observed sequence is Markovian. The latter assumption often fails 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.

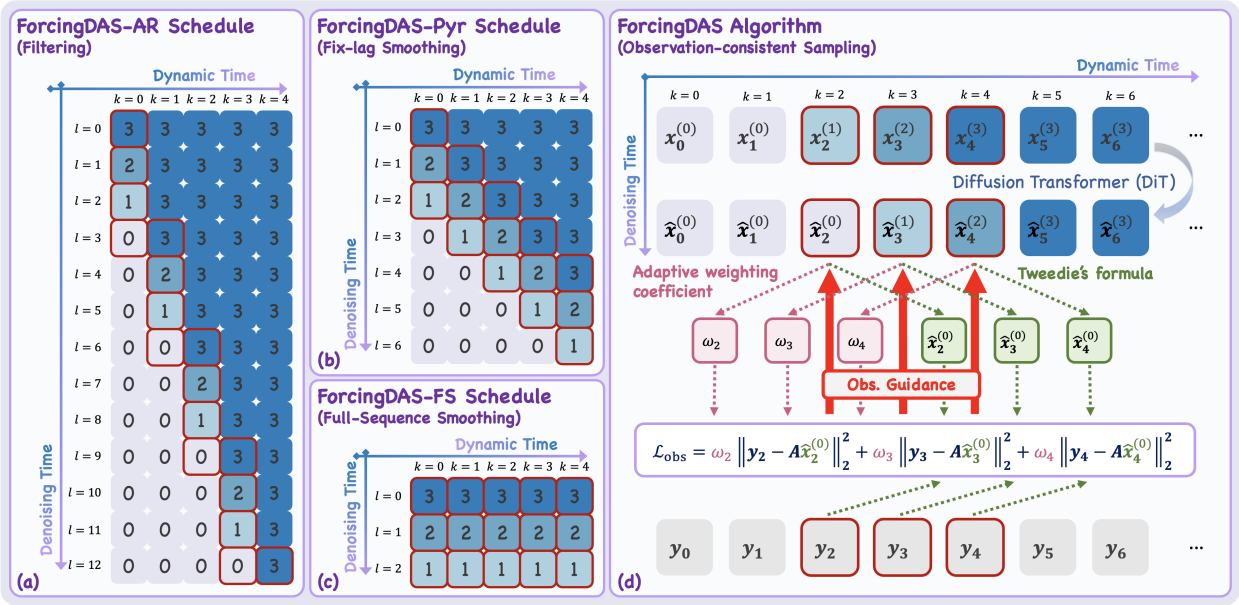

Figure 2. **ForcingDAS at a glance.** (a–c) The scheduling matrix $\boldsymbol{S}^{(u)}$ (rows: denoising iteration ℓ; columns: trajectory frame k; cells: diffusion step) selects the DA regime via the uncertainty scale u: autoregressive filtering ($u = T$), pyramid fixed-lag smoothing ($0 < u < T$), 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.

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.** Modeling full trajectories rather than per-step transitions allows the prior to capture dependencies on hidden degrees of freedom that a Markovian per-frame model misses, and joint denoising across frames mitigates the error accumulation issue that has limited autoregressive learned DA.
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. The largest gains appear on the non-Markovian benchmarks SEVIR-VIL and ERA5 (§4).

2. Background

2.1. Data Assimilation

The goal of DA is to estimate the state trajectory $\boldsymbol{x}_{1:K} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K)$ of the dynamical system (1) from partial observations $\boldsymbol{y}_{1:K} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K)$ 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 $t \in \{1, \ldots, T\}$ 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., 2020)) 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 $\boldsymbol{x}_{1:K}$:

$$\boldsymbol{x}_k^{(t_k)} = \sqrt{\bar{\alpha}_{t_k}}\,\boldsymbol{x}_k + \sqrt{1-\bar{\alpha}_{t_k}}\,\boldsymbol{\epsilon}_k, \tag{3}$$

with $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ for $k = 1, \ldots, K$. 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 $\boldsymbol{t} = (t_1, \ldots, t_K)$ and predicts the noise on each frame. The temporal backbone uses *causal attention*: frame k attends only to frames $k' \leq k$. At sampling time, a **scheduling matrix** $\boldsymbol{S} \in \{0, \ldots, T\}^{K \times L}$ 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 S3.

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

$\mathrm{Uniform}\{0,\ldots,T\}^K$, 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:

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

where p_{iid} is the standard $\mathrm{Uniform}\{1,\ldots,T\}^K$, 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 S2.

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., 2022). 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 by the inverse of this variance yields

$$\begin{aligned}\mathcal{L}_{\text{obs}} &= \sum_{k=1}^{K} w(t_k)\cdot\left\|\boldsymbol{y}_k-\mathcal{A}(\hat{\boldsymbol{x}}_k^{(0)})\right\|_2^2,\\ w(t_k) &= \frac{1}{\sqrt{\sigma_y^2+\gamma\,(1-\bar\alpha_{t_k})/\bar\alpha_{t_k}}},\end{aligned}\tag{5}$$

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

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

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 S4.

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

$$S_{k,\ell}^{(u)} = \mathrm{clip}\big(T-\ell+u\cdot(k-1),\;0,\;T\big),\tag{7}$$

for $k=1,\ldots,K$ and $\ell=0,\ldots,L-1$, where $L=T+u(K-1)$ and $\mathrm{clip}(\cdot,0,T)$ saturates 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) *causal smoothing* ($u = 0$, full-sequence), 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 S4 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 to long-range dependencies and non-Markovian observations*. By modeling the joint trajectory prior rather than a per-step transition, our ForcingDAS effectively captures the dynamics of low-dimensional projections and 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 S5.

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 S5.

4. Experiments

We evaluate ForcingDAS on three benchmarks. 2D Navier–Stokes vorticity (§4.1) is a Markovian, fully-observable PDE used to verify the framework; SEVIR-VIL precipitation nowcasting (§4.2) and ERA5 global atmospheric state estimation (§4.3) are the main empirical contributions, where the observed sequence is a low-dimensional projection of a higher-dimensional latent state, and joint trajectory modeling helps the most. Existing learned DA methods on ERA5 (Appendix S1.2, Table S1) all run in single-step filtering. A single trained model is evaluated in three regimes (filtering, fixed-lag smoothing with a 20-frame lag throughout, full-sequence smoothing) under our *with-context* protocol; the harder *cold-start* setting is in Appendix S7.

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 Markovian dynamics. Even in this setting, sparse-observation filtering is challenging because per-step error accumulates across long horizons. The main empirical content of the paper is on the non-Markovian benchmarks (§4.2, §4.3).

Setup, baselines, and metrics. We consider 2D incompressible Navier–Stokes vorticity at 128×128 resolution with $K = 30$ frames per trajectory, evaluated on 4 held-out trajectories. The headline operator is sparse pixel masking at 5% observed ratio (**SO-5%**) with additive Gaussian noise of $\sigma_y = 0.05$ in data space, with the first 10 frames clean as context; cold-start (no clean context; the model assimilates from observations alone) is in Appendix S7.1. We compare against FlowDAS (Chen et al., 2025), a learned filter trained on a per-step transition model; the learned smoother SDA (Rozet & Louppe, 2023) is reported in the cold-start appendix. We report NRMSE and the relative error of the radially-averaged kinetic-energy spectrum: for each frame we compute the radial spectrum $E(k)$ of the predicted

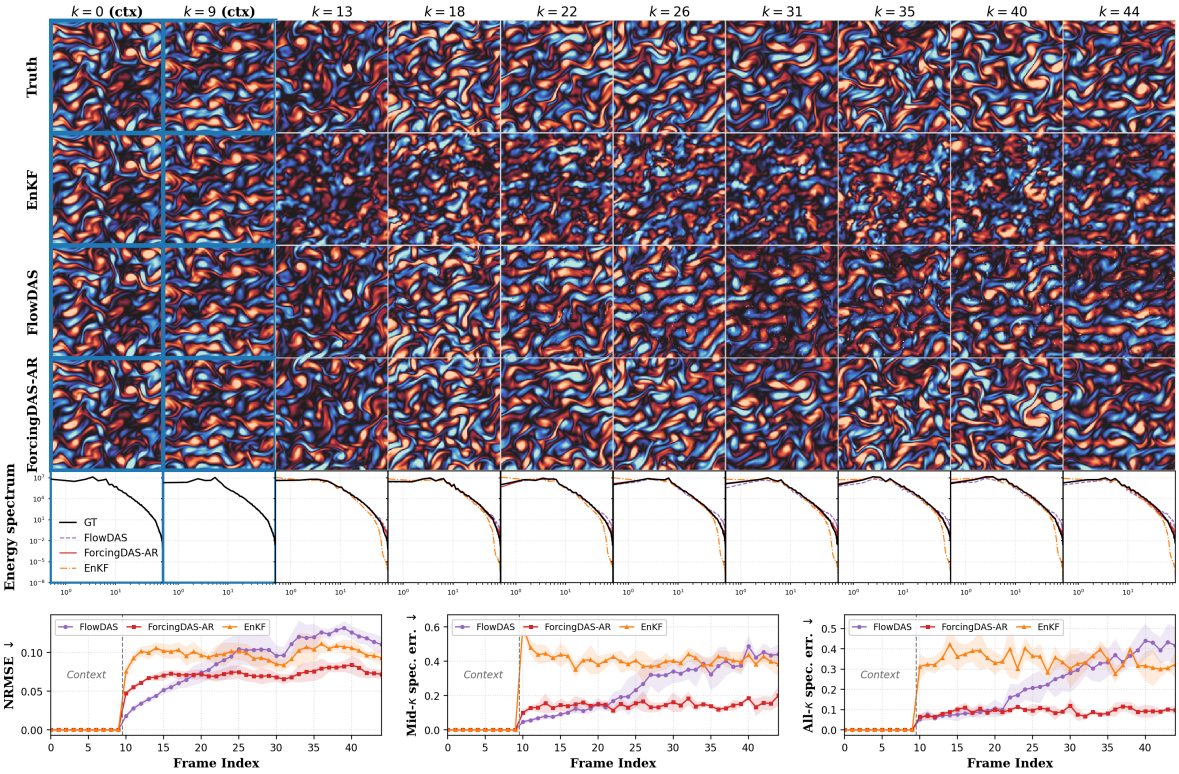

Figure 3. Per-frame *filtering* comparison on a representative NS trajectory under SO-5% with 10 clean context frames (blue-bordered columns). **Top four rows:** ground truth and predictions from the classical EnKF, the learned filter FlowDAS, and ForcingDAS-AR. **Fifth row:** per-frame radially-averaged kinetic-energy spectrum $E(k)$ on log-log axes. **Bottom:** per-frame NRMSE, mid-k, and all-k spectrum relative error (↓). The smoother comparison (EnKS-FS, SDA, ForcingDAS-FS) is deferred to Appendix S6.1.5.

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 ($\mathcal{K} = [0.5, 64)$). Dataset details and full hyperparameters are in Appendix S6.1.

Results. Table S6 reports SO-5% with-context numbers. ForcingDAS-AR improves over FlowDAS in the filtering regime on both NRMSE and the integrated all-k spectrum error; the fix-lag (Pyr) and full-sequence (FS) variants are similarly stable across the trajectory. FlowDAS, by contrast, accumulates per-step error on this sparse-observation setting and visibly diverges in late frames; Fig. 3 shows the qualitative breakdown on a representative trajectory.

4.2. Precipitation Nowcasting (SEVIR-VIL)

Setup. We use the Storm Event Imagery (SEVIR) dataset (Veillette et al., 2020), specifically the Vertically Integrated Liquid (VIL) product, a vertically integrated radar-derived measure of precipitation intensity. Frames are single-channel at 128×128, and we evaluate on 4 held-out trajectories with 6 clean context frames seeding a length-49 trajectory. We evaluate four observation operators with additive Gaussian noise $\sigma_y = 0.05$ in data space: sparse pixel masks at 10% and 20% observed ratios (**SO-10%**, **SO-20%**), and super-resolution operators at $2\times$ and $4\times$ (**SR2x**, **SR4x**). We additionally evaluate cold-start (no clean context) in Appendix S7.2, where ForcingDAS-FS matches SDA on NRMSE and outperforms it on every operator on CSI-160 and CSI-mean. Per-experiment details, including the equivalent σ_y in the standard 0–255 uint8 VIL convention, are in Appendix S6.2.

Baselines and metrics. We compare against FlowDAS (Chen et al., 2025) (a learned filter) and SDA (Rozet & Louppe, 2023) (a learned smoother). SEVIR has no operationally meaningful classical-DA baseline since the underlying VIL dynamics are not given by a known PDE. SDA does not accept clean context frames; its row in Table 1 therefore reflects the cold-start setting, while all other methods use the 6 context frames described above. We report NRMSE and the Critical Success Index (CSI) at threshold 160 (CSI-160) and the mean over thresholds $\{16, 74, 133, 160, 181, 219\}$ (CSI-mean) for precipitation detection skill; per-threshold CSI is in Appendix S6.2.5 and PSNR/SSIM in the Appendix.

Results. Table 1 reports NRMSE, CSI-160, and CSI-mean across the four observation operators. ForcingDAS-AR

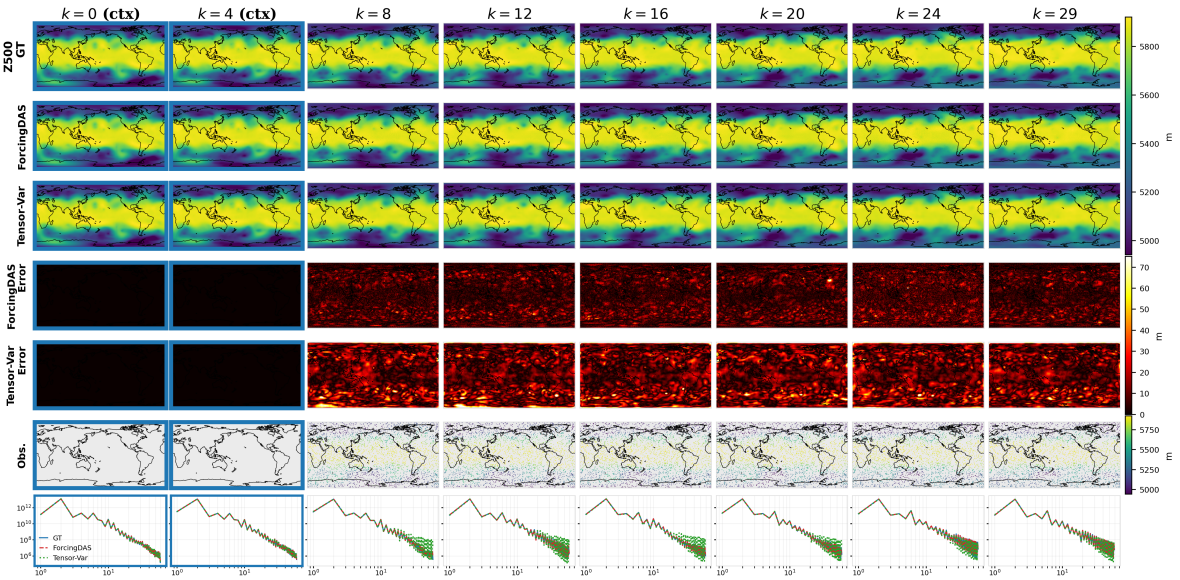

Figure 4. ERA5 SO-10% with-context assimilation, **Z500** (geopotential at 500 hPa) on a representative held-out trajectory. Rows (top to bottom): ground truth, ForcingDAS-Pyr prediction, TensorVar prediction, ForcingDAS-Pyr pixel-wise error, TensorVar pixel-wise error, sparse observation pattern, and the per-frame radially-averaged zonal-wavenumber spectrum overlaying predictions and ground truth. Columns are evenly-spaced frames $t \in \{0, 4, 8, 12, 16, 20, 24, 29\}$. The analogous panels for the other three variables (T850, U10, V10) are reported in Appendix S6.3.7 as Fig. S2, Fig. S3, and Fig. S4.

improves over FlowDAS in filtering, and ForcingDAS-FS improves over SDA in smoothing on every operator and metric; ForcingDAS-Pyr provides a fix-lag variant in the same range. Figure 1 visualizes per-frame VIL fields under FlowDAS and ForcingDAS-Pyr against the ground truth on a representative trajectory.

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 ground-truth atmospheric state and run the same system under both filtering (real-time forecasting) and smoothing (retrospective reanalysis), directly testing the unification claim. Data come from WeatherBench2 (Rasp et al., 2020) at $\sim$1.5° on a 240×121 grid (6-hourly cadence); we train on 1979–2015 and evaluate on 4 held-out length-30 trajectories from 2016. Four variables span upper-air dynamics, thermodynamics, and surface wind: **Z500** (geopotential at 500 hPa), **T850** (temperature at 850 hPa), **U10** and **V10** (10 m zonal and meridional wind), stacked as input ($C = 4$, shape $4 \times 121 \times 240$) with per-variable z-score normalization. See Appendix S6.3.2 for variable descriptions and Appendix S1.2 for the broader landscape (Table S1).

Observation setup. At test time we simulate sparse observations with a shared random pixel mask (same mask across all 4 channels) and additive Gaussian noise. The headline setting is 10% observed at $\sigma_y = 0.05$ in data space (**SO-10%**), with 6 clean context frames followed by $K - 6$ assimilated frames; cold-start (no clean context) is in Appendix S7.3, where ForcingDAS-FS retains its advantage over 3D-Var and TensorVar. Equivalent σ_y in raw physical units is in Appendix S6.3.

Baselines and metrics. We compare ForcingDAS against **3D-Var** (classical filter; independent variational analysis per step with diagonal background covariance $\boldsymbol{B}$ and climatological background $\boldsymbol{x}_b = \mathbf{0}$), **FlowDAS** (Chen et al., 2025) (learned filter; per-step posterior sampling driven by a 29M-parameter 2D UNet forecast model), and **Tensor-Var** (Yang et al., 2025) (learned 4D-Var; latent-space linear dynamics with a transformer inverse observation network, solved as a closed-form quadratic program over the window; adapted from its native 64×32 grid to our 240×120). All baselines use the same observation operator, noise level, and data split (Appendix S8). Per variable, we report latitude-weighted NRMSE (RMSE in z-score space, per WeatherBench2 (Rasp et al., 2020)), the anomaly correlation coefficient (ACC, latitude-weighted Pearson correlation against the 1990–2019 ERA5 climatology), and latitude-weighted Bias (signed mean error; closer to zero is better). All metrics use cosine-of-latitude weights; full definitions are in Appendix S6.3.4,

Table 1. SEVIR VIL nowcasting with 6 clean context frames seeding a length-49 trajectory: NRMSE (↓), CSI-160 (↑), and CSI-mean (↑) over thresholds $\{16, 74, 133, 160, 181, 219\}$. Averaged over 4 held-out trajectories. Methods are grouped by inference regime: *Filtering* (FlowDAS, ForcingDAS-AR), *Fix-lag Smoothing* (ForcingDAS-Pyr), and *Smoothing* (SDA, ForcingDAS-FS); best in *Filtering* and best in *Smoothing* (Fix-lag pooled with Smoothing) per column is in **bold**. Per-threshold CSI is in Table S8; cold-start results are in Appendix S7.2.

	SO-10%			SO-20%			SR2x			SR4x		
Method	NRMSE	CSI-160	CSI-m	NRMSE	CSI-160	CSI-m	NRMSE	CSI-160	CSI-m	NRMSE	CSI-160	CSI-m
Filtering												
FlowDAS	0.453	0.464	0.556	0.382	0.547	0.603	0.200	0.670	0.721	0.371	0.477	0.581
ForcingDAS-AR	**0.291**	**0.533**	**0.631**	**0.224**	**0.622**	**0.699**	**0.169**	**0.675**	**0.747**	**0.301**	**0.522**	**0.607**
Fix-lag Smoothing												
ForcingDAS-Pyr	0.289	0.533	0.630	0.228	0.618	0.697	0.169	0.672	0.743	0.295	0.517	0.606
Smoothing												
SDA	0.291	0.534	0.633	0.216	0.617	0.697	0.184	0.648	0.720	0.307	0.511	0.611
ForcingDAS-FS	**0.256**	**0.559**	**0.650**	**0.199**	**0.636**	**0.714**	**0.150**	**0.689**	**0.753**	**0.255**	**0.540**	**0.636**

Table 2. ERA5 SO-10% assimilation with 6 clean context frames: latitude-weighted NRMSE (↓, normalized units), anomaly correlation coefficient (ACC, ↑), and latitude-weighted Bias (signed mean error; closer to 0 is better) per variable. Averaged over 4 held-out length-30 trajectories from 2016. Methods are grouped by inference regime: *Filtering* (3D-Var, FlowDAS, ForcingDAS-AR), *Fix-lag Smoothing* (ForcingDAS-Pyr), and *Smoothing* (TensorVar, ForcingDAS-FS); best in *Filtering* and best in *Smoothing* (Fix-lag pooled with Smoothing) per column is in **bold**. Cold-start results are in Appendix S7.3.

	Z500			T850			U10			V10		
Method	NRMSE	ACC	Bias	NRMSE	ACC	Bias	NRMSE	ACC	Bias	NRMSE	ACC	Bias
Filtering												
3D-Var	0.059	0.971	**0.000**	0.094	0.906	**0.000**	0.392	0.833	0.003	0.449	0.846	-0.007
FlowDAS	0.031	0.992	-0.007	0.062	0.957	-0.007	0.175	0.966	-0.016	0.213	0.964	-0.022
ForcingDAS-AR	**0.023**	**0.996**	-0.001	**0.047**	**0.976**	-0.003	**0.133**	**0.980**	**-0.002**	**0.164**	**0.979**	**-0.006**
Fix-lag Smoothing												
ForcingDAS-Pyr	0.023	0.996	-0.001	0.047	0.976	-0.003	0.133	0.980	-0.002	0.163	0.979	-0.006
Smoothing												
TensorVar	0.037	0.989	-0.009	0.059	0.962	-0.008	0.170	0.967	**0.000**	0.211	0.964	-0.031
ForcingDAS-FS	**0.019**	**0.997**	**0.000**	**0.042**	**0.981**	**-0.001**	**0.126**	**0.982**	-0.001	**0.155**	**0.981**	**-0.006**

and per-variable zonal-wavenumber spectrum error appears in the last row of Fig. 4 and the analogous panels in Appendix S6.3.7.

Results. Table 2 reports per-variable NRMSE, ACC, and Bias under SO-10% with 6 clean context frames; Fig. 4 shows ForcingDAS against TensorVar on a representative Z500 trajectory, with analogous panels for the other variables in Appendix S6.3.7. 3D-Var attains high ACC on smooth synoptic-scale fields (Z500, T850) since its Gaussian spatial correlation in $\boldsymbol{B}$ already interpolates large-scale patterns from 10% pixel coverage, but cannot recover fine-scale structure: NRMSE is 2–7× higher than ForcingDAS across all variables, with the largest gap on surface winds. Both TensorVar and ForcingDAS use learned temporal dynamics, and ForcingDAS attains the lowest NRMSE on every variable.

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, compute that scales with the active window for joint-denoising regimes, and a coarse ERA5 setup relative to operational systems. See Appendix S9 for the full discussion.

References

Andry, G., Lewin, S., Rozet, F., Rochman, O., Mangeleer, V., Pirlet, M., Faulx, E., Grégoire, M., and Louppe, G. Appa: Bending weather dynamics with latent diffusion models for global data assimilation. *arXiv*, 2025. doi: 10.48550/arxiv.2504.18720.

Asch, M., Bocquet, M., and Nodet, M. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.

Banerjee, A., van Dinther, Y., and Vossepoel, F. C. On parameter bias in earthquake sequence models using data assimilation. *Nonlinear Processes in Geophysics*, 30(2):101–115, 2023.

Bickel, P., Li, B., and Bengtsson, T. 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. 318–330. Institute of Mathematical Statistics, 2008.

Bocquet, M., Elbern, H., Eskes, H., Hirtl, M., Žabkar, R., Carmichael, G., Flemming, J., Inness, A., Pagowski, M., Pérez Camaño, J., 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.

Brunton, S. L. and Noack, B. R. Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, 67(5):050801, 2015.

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

Carrassi, A., Bocquet, M., Bertino, L., and Evensen, G. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, 9(5):e535, 2018.

Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., and Sitzmann, V. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024a.

Chen, K., Bai, L., Ling, F., Ye, P., Chen, T., Luo, J.-J., Chen, H., Xiao, Y., Chen, K., Han, T., and Ouyang, W. Towards an end-to-end artificial intelligence driven global weather forecasting system. *arXiv preprint arXiv:2312.12462*, 2024b.

Chen, K., Ye, P., Chen, H., Han, T., Ouyang, W., Chen, T., BAI, L., et al. Fnp: Fourier neural processes for arbitrary-resolution data assimilation. *Advances in Neural Information Processing Systems*, 37:137847–137872, 2024c.

Chen, S., Jia, Y., Qu, Q., Sun, H., and Fessler, J. A. Flowdas: A stochastic interpolant-based framework for data assimilation. *arXiv preprint arXiv:2501.16642*, 2025.

Chen, Y., Goldstein, M., Hua, M., Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Probabilistic forecasting with stochastic interpolants and Föllmer processes. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (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 2024d. URL https://proceedings.mlr.press/v235/chen24n.html.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Courtier, P., Thépaut, J.-N., and Hollingsworth, A. A strategy for operational implementation of 4d-var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994.

Courtier, P., Andersson, E., Heckley, W., Vasiljevic, D., Hamrud, M., Hollingsworth, A., Rabier, F., Fisher, M., and Pailleux, J. The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: Formulation. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1783–1807, 1998.

Cummings, J. A. Operational multivariate ocean data assimilation. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613):3583–3604, 2005.

Cummings, J. A. and Smedstad, O. M. Variational data assimilation for the global ocean. In *Data assimilation for atmospheric, oceanic and hydrologic applications (Vol. II)*, pp. 303–343. Springer, 2013.

Duraisamy, K., Iaccarino, G., and Xiao, H. Turbulence modeling in the age of data. *Annual review of fluid mechanics*, 51(1):357–377, 2019.

Evensen, G. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53: 343–367, 2003.

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

Fletcher, S. J. *Data assimilation for the geosciences: From theory to application*. Elsevier, 2017.

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

Geer, A. J., Lonitz, K., Weston, P., Kazumori, M., Okamoto, K., Zhu, Y., Liu, E. H., Collard, A., Bell, W., Migliorini, S., et al. All-sky satellite data assimilation at operational weather forecasting centres. *Quarterly Journal of the Royal Meteorological Society*, 144(713):1191–1217, 2018.

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

Gordon, N. J., Salmond, D. J., and Smith, A. F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pp. 107–113. IET, 1993.

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

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730): 1999–2049, 2020.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Huang, L., Gianinazzi, L., Yu, Y., Dueben, P. D., and Hoefler, T. DiffDA: a diffusion model for weather-scale data assimilation. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (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. URL https://proceedings.mlr.press/v235/huang24h.html.

Huang, L. et al. Physically consistent global atmospheric data assimilation with machine learning in a latent space. *arXiv preprint arXiv:2502.02884*, 2025a.

Huang, X., Li, Z., He, G., Zhou, M., and Shechtman, E. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025b.

Kálmán, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

Lahoz, W., Khattatov, B., and Menard, R. *Data assimilation*. Springer, 2010.

Law, K., Stuart, A., and Zygalakis, K. Data assimilation. *Cham, Switzerland: Springer*, 214:52, 2015.

Mojgani, R., Chattopadhyay, A., and Hassanzadeh, P. 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.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Rabier, F. Overview of global data assimilation developments in numerical weather-prediction centres. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613):3215–3233, 2005.

Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., and Thuerey, N. WeatherBench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

Reichle, R. H. Data assimilation methods in the earth sciences. *Advances in water resources*, 31(11):1411–1418, 2008.

Rozet, F. and Louppe, G. Score-based data assimilation. *Advances in Neural Information Processing Systems*, 36: 40521–40541, 2023.

Ruhe, D., Heek, J., Salimans, T., and Hoogeboom, E. Rolling diffusion models. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.09470.

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

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Song, K., Chen, B., Simchowitz, M., Du, Y., Tedrake, R., and Sitzmann, V. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

Taira, Y., Sagara, S., and Oya, M. Model-based motion control for underwater vehicle-manipulator systems with one of the three types of servo subsystems. *Artificial life and robotics*, 25:133–148, 2020.

Talagrand, O. and Courtier, P. Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113(478):1311–1328, 1987.

Van Leeuwen, P. J. Particle filtering in geophysical systems. *Monthly Weather Review*, 137(12):4089–4114, 2009.

Veillette, M., Samsi, S., and Mattioli, C. 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.

Werner, M. J., Ide, K., and Sornette, D. Earthquake forecasting based on data assimilation: Sequential monte carlo methods for renewal processes, 2009. URL https://arxiv.org/abs/0908.1516.

Xiang, Y., Jin, W., Dong, H., Bai, M., Fang, Z., Zhao, P., Sun, H., Thambiratnam, K., Zhang, Q., and Huang, X. ADAF: An artificial intelligence data assimilation framework for weather forecasting. *arXiv preprint arXiv:2411.16807*, 2024.

Xiao, Y., Bai, L., Xue, W., Chen, K., Han, T., and Ouyang, W. FengWu-4DVar: Coupling the data-driven weather forecasting model with 4D variational assimilation. *arXiv preprint arXiv:2312.12455*, 2024.

Xu, X., Zhong, X., Zhang, H., Qi, Y., Chen, L., and Li, X. FuXi-DA: A generalized deep learning data assimilation framework for assimilating satellite observations. *npj Climate and Atmospheric Science*, 8(1):156, 2025.

Yang, Y., Gao, X., Salinas, D., and Cantu-Paz, E. Tensor-var: Variational data assimilation in tensor feature space with deep kernel learning. *arXiv preprint arXiv:2501.13312*, 2025.

Appendices

S1. Related works

This appendix collects related work along two axes: (i) per-frame-noise diffusion methods that share the prior structure of Diffusion Forcing (§S1.1); and (ii) classical and learned data assimilation methods (§S1.2).

S1.1. Per-Frame-Noise Diffusion Methods

ForcingDAS uses Diffusion Forcing (Chen et al., 2024a) as its trajectory prior. We position DF and the related per-frame-noise diffusion methods that have appeared since.

Rolling Diffusion (Ruhe et al., 2024). Rolling Diffusion sweeps a window across a sequence with frames at monotonically decreasing 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., 2025b). 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 DPS-style guidance, and no filtering / fixed-lag / smoothing distinction. The DA reformulation, the scheduling-matrix family $\boldsymbol{S}^{(u)}$, and the noise-level-aware observation guidance are introduced by ForcingDAS.

S1.2. Classical and Learned Data Assimilation

We organize related DA methods along two axes: classical (variational and ensemble) and learned (per-frame transitions and trajectory priors).

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 §S8.

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 (Kálmán, 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).

Learned filters: per-frame transition priors (FlowDAS, TensorVar). FlowDAS (Chen et al., 2025) learns a single-step stochastic interpolant transition and runs DPS-style guidance frame-by-frame, giving an online filter. Tensor-Var (Yang et al., 2025) learns a linear latent dynamics model coupled with a transformer-based observation network and solves the resulting variational problem in closed form over a short window. Both commit to a single regime by training: FlowDAS is a filter, Tensor-Var operates over a small fixed window. Because the prior is tied to a per-step transition, both inherit the assumption that the observed sequence is Markovian, which fails on partial observations of a higher-dimensional latent state (§4.2, §4.3).

Learned smoothers: trajectory priors (SDA). Score-Based Data Assimilation (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 DF to expose filtering and fixed-lag smoothing as well as full-sequence smoothing from a single trained model.

Learned DA on ERA5. ERA5 has become the de facto evaluation benchmark for learned global atmospheric DA, and several recent systems run filtering 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., 2024b) 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 DL 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 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** (Huang et al., 2025a) 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., 2024) 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 autodiff through the learned forward model, replacing the hand-coded operational adjoint.

All listed methods commit to a single regime, single-step or cyclic filtering, and none exposes the filtering–smoothing spectrum from a single trained model. ForcingDAS is, to our knowledge, the first to do so on ERA5.

Table S1. Landscape of learned DA methods on ERA5. All listed methods perform single-step or cyclic *filtering*; none exposes the filtering–smoothing spectrum. "Vars" counts total input channels (upper-air variables $\times$ pressure levels $+$ surface variables). "Obs" indicates 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 (Huang et al., 2024)	0.25°	721×1440	82	$\geq$37M†	Simul. columns	Filtering
FengWu-Adas (Chen et al., 2024b)	0.25°	721×1440	69	N/R	Simul. + GDAS	Cyclic filtering
FuXi-DA (Xu et al., 2025)	0.25°	721×1440	70	N/R	FY-4B sat.	Filtering
Appa (Andry et al., 2025)	0.25°	721×1440	71	565M	Arbitrary (0-shot)	Filtering*
L4DVar (Huang et al., 2025a)	1.4°	128×256	69	N/R	GDAS real	Latent 4D-Var
ADAF (Xiang et al., 2024)	0.05°	512×1280	4	N/R	Sfc + GOES-16	Filtering (CONUS)
FengWu-4DVar (Xiao et al., 2024)	0.25°	721×1440	69	N/R	Simulated	4D-Var (AI adj.)
ForcingDAS (ours)	1.5°	240×121	4	$\sim$30M	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; adapted diffusion variant likely larger (not reported).

S2. Effect of Causality-Aware Training on Probabilistic Forecasting

This appendix 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 unconditional generation, so probabilistic forecasting is the cleanest setting in which to isolate it. Showing that CAT improves forecasting on each of the three benchmarks then carries over to the DA results in §4.

S2.1. 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 (matching §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 S2–S3; on ERA5 we report $\rho = 0.25$. The remaining mixture parameters $(\rho_c^\star, C_{\max})$ follow the values in Appendix S5.

This is the cleanest possible ablation of CAT: any difference in forecasting quality is attributable to the training noise distribution alone.

Metrics. We report two metrics per dataset: the Continuous Ranked Probability Score (CRPS, §S2.2) 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 S6.3.4).

S2.2. 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

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} \big(F(z) - \mathbf{1}\{y \le z\}\big)^2 \, \mathrm{d}z, \tag{S1}$$

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

$$\widehat{\mathrm{CRPS}}\Big(\{x^{(i)}\}_{i=1}^{M}, y\Big) = \frac{1}{M}\sum_{i=1}^{M}\big|x^{(i)} - y\big| - \frac{1}{2M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\big|x^{(i)} - x^{(j)}\big|, \tag{S2}$$

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. (S2) 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. (S2) 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. (S14), so the reported per-variable CRPS is a latitude-weighted mean exactly analogous to the latitude-weighted NRMSE of Eq. (S15).

S2.3. Results

Table S2. 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

Table S3. 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 S2.

	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 S4. 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 (per-channel std ≈ 1); 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**

This subsection expands on the motivation for CAT (§3.1) and reports its effect on probabilistic forecasting under the protocol above.

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 to use a clean past as a reliable conditioning signal; training with i.i.d. noise levels teaches the model to tolerate noisy context, which is the wrong inductive bias when the inference path always provides a clean 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 S2–S4 report CRPS and ensemble-mean NRMSE per lead time on the three benchmarks. CAT is most effective on ERA5 (Table S4): 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, $0.053 \rightarrow 0.027$; T850, $0.058 \rightarrow 0.038$). On SEVIR (Table S3), $\rho = 0.75$ 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 S2), 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 signal that AR inference relies on is therefore no more informative than what an i.i.d.-trained network already learns to use, and the per-step transition is already learned well by standard DF training, leaving little headroom for CAT to improve. 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 with noisy future is exactly the regime in which the model needs to be trained, and matching it through CAT translates into clear forecasting gains. The fact 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.

S3. Diffusion Forcing: Background, Recap, and Demarcation

This appendix recaps Diffusion Forcing (DF) (Chen et al., 2024a) (the per-frame-noise diffusion model that ForcingDAS adopts as its trajectory prior) and marks which components ForcingDAS inherits and which are introduced in this work. We follow the ForcingDAS notation of §2.2: $k = 1, \dots, K$ is the frame index, $t = 1, \dots, T$ is the diffusion-step index, t_k is the per-frame diffusion step, and $\boldsymbol{t} = (t_1, \dots, t_K)$. The original DF paper instead uses T for sequence length (our K), K for diffusion total (our T), and k_t for t_k.

S3.1. 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 $\boldsymbol{x}_{1:K} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_K) \sim q$, draw independent noise levels $t_k \in \{0, \dots, T\}$ and Gaussian noise $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and form

$$\boldsymbol{x}_k^{(t_k)} = \sqrt{\bar{\alpha}_{t_k}}\, \boldsymbol{x}_k + \sqrt{1 - \bar{\alpha}_{t_k}}\, \boldsymbol{\epsilon}_k, \quad k = 1, \dots, K, \tag{S3}$$

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 unified "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 Markovianly 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); ForcingDAS uses a causal-attention transformer instead (§S3.4). Both choices are length-invariant in K.

S3.2. Training Procedure

Standard DF training samples $t_k \overset{\text{i.i.d.}}{\sim} \mathrm{Uniform}\{0, \dots, T\}$ across frames, corrupts $\boldsymbol{x}_{1:K}$ via Eq. (S3), and minimizes the per-frame score-matching loss

$$\mathcal{L}_{\mathrm{DF}}(\boldsymbol{\theta}) \;=\; \mathbb{E}_{\boldsymbol{x}_{1:K},\, \boldsymbol{t},\, \boldsymbol{\epsilon}_{1:K}} \sum_{k=1}^{K} \left\| \boldsymbol{\epsilon}_k - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\big(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t}\big)_k \right\|^2, \tag{S4}$$

The defining feature is the per-frame independence of $\boldsymbol{t}$: it exposes the network to all T^K 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 full training pseudocode in our notation is given by Algorithm S1.

S3.3. Sampling Procedure

DF sampling is controlled by a *scheduling matrix* $\boldsymbol{S} \in \{0, \dots, T\}^{K \times L}$ (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., 2020) step takes each $k \in \mathcal{K}_{\text{active}}$ from $t_k = S_{k,\ell}$ to $t'_k = S_{k,\ell+1}$:

$$\boldsymbol{x}_k^{(t'_k)} = \sqrt{\bar{\alpha}_{t'_k}}\, \hat{\boldsymbol{x}}_k^{(0)} + \sqrt{1 - \bar{\alpha}_{t'_k} - \sigma_{t_k}^2}\, \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:K}^{(\boldsymbol{t})}, \boldsymbol{t})_k + \sigma_{t_k}\, \boldsymbol{z}_k, \quad \boldsymbol{z}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{S5}$$

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

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); all of these ForcingDAS recasts as DA regimes (filtering, full-sequence smoothing, fixed-lag smoothing) and embeds into the continuous 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. (S5). 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 $k \leq k'$, the gradient flows backwards in time and refines past frames, the "long-horizon guidance" mechanism that ForcingDAS reuses for observation guidance (§3.1, Appendix S4). The Monte Carlo Guidance variant of (Chen et al., 2024a), which averages gradients over multiple future roll-outs, is unnecessary for DA and not used. The full sampling pseudocode is given by Algorithm S2.

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

Require: $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$, scheduling matrix $\boldsymbol{S} \in \{0, \ldots, T\}^{K \times L}$, optional cost $c(\cdot)$, scale $\zeta \geq 0$
$\boldsymbol{x}_k^{(T)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $k = 1, \ldots, K$
for $\ell = 0, \ldots, L-1$ **do**
  $t_k \leftarrow S_{k,\ell}$, $t'_k \leftarrow S_{k,\ell+1}$; $\mathcal{K}_{\text{active}} \leftarrow \{k : t'_k < t_k\}$
  Apply Eq. (S5) to each $k \in \mathcal{K}_{\text{active}}$
end for
Return $\boldsymbol{x}_{1:K}^{(0)}$

S3.4. Model Architecture and Training Setup

Original DF instantiation. The original DF paper (Chen et al., 2024a) instantiates $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ as a causal recurrent network: per-frame features are produced by a 2D U-Net (for video data, e.g., DMLab and Minecraft) or a residual MLP (for low-dimensional planning and time-series data); the resulting embeddings are fed through a GRU along the temporal axis to produce a hidden state $\boldsymbol{z}_k$ that summarizes the noisy history; an observation head finally maps $\boldsymbol{z}_k$ to the noise prediction $\boldsymbol{\epsilon}_{\boldsymbol{\theta},k}$. The latent channel widths and parameter counts reported in (Chen et al., 2024a) are 16 channels / ≈24M parameters for DMLab, 32 channels / ≈36M parameters for Minecraft, and ≈4.3M parameters for D4RL maze planning. Training uses fp16 mixed precision with $T = 1000$ diffusion steps; sampling uses 100 DDIM (Song et al., 2020) steps on video and 50 on non-video domains. Video models are trained for 50K steps with batch size 8×16 on 8 A100 GPUs (≈12h to convergence), while the planning, time-series, and imitation-learning models fit on a single 11GB 2080 Ti GPU in 4–8 hours. Two implementation knobs of (Chen et al., 2024a) are worth singling out: *Fused SNR reweighting*, which re-derives min-SNR-style loss weighting for the per-frame setting by combining the SNR of the current frame with an exponentially-decaying running mean of past-frame SNRs (essential for video-prediction convergence); and *frame stacking*, which packs F consecutive raw frames into one super-frame ($F = 4$ on DMLab, $F = 8$ on Minecraft, $F = 10$ on D4RL maze), reducing the temporal dimension fed to the RNN. The prediction parameterization is task-dependent: v-parameterization for video, $\boldsymbol{x}_0$-parameterization for planning and imitation learning, and $\boldsymbol{\epsilon}$-parameterization for time series. Appendix B of (Chen et al., 2024a) sketches a transformer extension that uses per-frame noise levels (rather than an explicit causal mask) to control the directionality of information flow, an idea also pursued at scale by DFoT (Song et al., 2025).

ForcingDAS instantiation. ForcingDAS replaces the RNN+U-Net stack with a 3D Diffusion Transformer (DiT-3D) (Peebles & Xie, 2023; Chen et al., 2024a) backbone (full architectural details in §S5.1): each frame is patchified into spatial tokens, full spatial self-attention runs within each frame, causal temporal self-attention runs across frames so that frame k attends only to frames $k' \leq k$, and the per-frame diffusion step t_k is injected through adaptive layernorm conditioned on a sinusoidal embedding of t_k. The architectural motivation is that the scientific dynamical fields targeted by ForcingDAS (2D Navier–Stokes vorticity, SEVIR radar, ERA5 weather) have broadband spatial spectra and rapidly evolving structure for which transformer-style global attention is empirically a stronger inductive bias than the local convolutional / recurrent backbone of (Chen et al., 2024a). The training setup itself is broadly inherited from DF: the per-frame Gaussian forward process of Eq. (S3), $\boldsymbol{\epsilon}$-parameterization, a cosine variance schedule, $T = 1000$ diffusion steps, AdamW optimization with mixed-precision training, and 100-step DDIM sampling; per-dataset hyperparameters (patch size, hidden width, number of layers, batch size, learning rate) are listed in the per-experiment appendices (§S6). Frame stacking is retained for the high-frame-rate benchmarks (NS, SEVIR) but adapted to the channel-stacking convention of the DiT input (§S5.2).

S4. 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 $k \leq k'$ and zero otherwise (Eq. (S6) below), guidance at a future frame k' does produce a gradient on frame k, but an *indirect* one attenuated by the depth of attention and nonlinearities it traverses. Consequently, ForcingDAS sits *between* pure filtering and fully bidirectional smoothing: even in its strongest mode ($u = 0$), future observations influence past states only through this backward gradient, strictly stronger than filtering but weaker than a hypothetical model with bidirectional forward-pass attention. We give the formal chain-rule decomposition below (Remark 1) and quantify the asymmetry empirically in §4.

Forward direction (strong): 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 strong and direct information pathway. This is the *filtering* channel.

Backward direction (weak): 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 $k \leq k'$, these backward gradients are non-zero, but *indirect*: they pass through softmax Jacobians and nonlinearities, and hence attenuated relative to the forward pathway. 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

$$\frac{\partial\, g_{k'}}{\partial\, \boldsymbol{x}_k^{(t_k)}} \begin{cases} \neq \mathbf{0} & \text{if } k \leq k', \\ = \mathbf{0} & \text{if } k > k'. \end{cases} \tag{S6}$$

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

$$\nabla_{\boldsymbol{x}_k^{(t_k)}} \|\boldsymbol{y}_{k'} - \mathcal{A}(g_{k'})\|^2 = \underbrace{-2\, \boldsymbol{J}_{\mathcal{A}}^{\top} (\boldsymbol{y}_{k'} - \mathcal{A}(g_{k'}))}_{\text{observation residual signal}} \cdot \underbrace{\frac{\partial\, g_{k'}}{\partial\, \boldsymbol{x}_k^{(t_k)}}}_{\text{cross-frame Jacobian}} , \tag{S7}$$

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. (S6), but attenuated by the depth of the causal path from frame k to k' through the network's attention and nonlinear layers.

S5. Implementation Details and Pseudocode

Algorithm S3 ForcingDAS Training

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

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

Algorithm S4 ForcingDAS Sampling (Inference)

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

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

This appendix expands on the implementation summary in §3.2: it gives the network backbone, additional design choices, and full pseudocode for the training and inference procedures referenced from the main text.

S5.1. Network architecture

The denoising network $\boldsymbol{\epsilon}_\theta$ 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 full-image within each frame, while temporal attention is restricted to the causal mask: frame k attends only to frames $k' \leq k$. The same backbone is used across all three benchmarks; per-dataset choices (patch size, hidden width, number of layers) are listed in the per-experiment appendices (§S6).

S5.2. Other key design choices

Frame stacking. For high-frame-rate benchmarks (NS, SEVIR), we stack F consecutive frames into a single super-frame with $C \cdot F$ channels, reducing the temporal dimension fed to the network from T_{raw} to $K = T_{\text{raw}}/F$. The per-super-frame noise-level structure is preserved.

Noise clipping. For z-score-normalized weather data with 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. When the test trajectory is longer than the training horizon K, we maintain a sliding window of length K: only frames inside the current window are fed to the network at each iteration, and the window advances by the chunk size $\lceil K/u \rceil$ for autoregressive sampling or stays fixed for full-sequence sampling.

S5.3. Pseudocode

Algorithm S3 consolidates the training procedure (sample CAT-mixture diffusion-step vector, corrupt, score-match). Algorithm S4 consolidates the inference procedure (read scheduling matrix, identify active frames, compute reweighted observation gradient, apply DDIM-with-correction step).

S6. Per-Experiment Details

This appendix collects the additional details that did not fit in the main text for each of the three experimental systems studied in §4: (i) further dataset background and pointers to related work; (ii) the precise observation model and the conversion between the data-space noise level used at inference time and its raw-physical-units counterpart; (iii) sampling-time hyperparameters used to produce the headline numbers; and (iv) supplementary visualizations and results.

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 (§S6.1), $[0, 1]$-scaled VIL reflectivity for SEVIR (§S6.2), and per-channel z-score-normalized fields for ERA5 (§S6.3). 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 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

$$\sigma_y^{\text{raw}} \;=\; b \cdot \sigma_y^{\text{data}}, \qquad b \;=\; \begin{cases} x_{\max} - x_{\min} & \text{(min–max scaling, NS \& SEVIR)}, \\ \sigma_c^{\text{clim}} & \text{(per-channel } z\text{-score, ERA5)}. \end{cases} \tag{S8}$$

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. (S8) is therefore an exact, lossless restatement, not an approximation, and the per-experiment subsubsections below tabulate the resulting σ_y^{raw} in physical units.

S6.1. Navier–Stokes Vorticity

S6.1.1. 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., 2024d; Rozet & Louppe, 2023):

$$\mathrm{d}\omega + (\boldsymbol{v} \cdot \nabla\omega)\,\mathrm{d}t = \nu\,\Delta\omega\,\mathrm{d}t - \alpha\,\omega\,\mathrm{d}t + \varepsilon\,\mathrm{d}\eta, \tag{S9}$$

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 $\{W_i(t)\}_{i=1}^{8}$:

$$\begin{aligned} \eta(t, x, y) = {} & W_1(t)\sin(6x) + W_2(t)\cos(7x) + W_3(t)\sin\big(5(x{+}y)\big) + W_4(t)\cos\big(8(x{+}y)\big) \\ & + W_5(t)\cos(6x) + W_6(t)\sin(7x) + W_7(t)\cos\big(5(x{+}y)\big) + W_8(t)\sin\big(8(x{+}y)\big). \end{aligned} \tag{S10}$$

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. We use $K \in \{30, 50\}$ frames per trajectory and evaluate on 4 held-out test trajectories. Because the underlying PDE is fully specified, this experiment also admits strong physics-based DA references (the Ensemble Kalman Filter and Smoother), whose implementation are described in Appendix S8.4.

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.

S6.1.2. Observation model and noise calibration

Operators. For NS we evaluate the 5% sparse observation operator (cf. Setup in §4.1): random binary pixel mask retaining 5% of grid points; the mask is a fixed (seed-controlled) sample.

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. (S8) with $b = 36.58$, this is exactly equivalent to applying the same operator (same mask / same down-sampling kernel) to the *raw* vorticity field and adding Gaussian noise of standard deviation

$$\sigma_y^{\text{raw}} \;=\; 0.05 \times 36.58 \;=\; 1.829 \;\; \text{(rad/s)} \;\;\approx\; 0.60 \times \sigma^{\text{raw}}. \tag{S11}$$

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.

S6.1.3. 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.

S6.1.4. 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 DPS guidance scale ζ. Table S5 reports the values used to produce the headline numbers in the main text; (γ, ζ) is selected by a small grid search on a single held-out trajectory before evaluating on the 4-trajectory test set.

Table S5. NS sampling hyperparameters (DiT-B backbone, 100 DDIM steps, $\sigma_y^{\text{data}} = 0.05$). AR = autoregressive (filter), Pyr = pyramid (fixed-lag), FS = full-sequence (smoothing). FS configurations are selected on the spec_err criterion (best radially-averaged spectrum over the held-out trajectory).

Operator	Schedule	T (frames)	Context	γ	ζ
SO-5%	AR	30	0	0.05	4.0
SO-5%	Pyr	30	0	0.05	4.0
SO-5%	FS	30	0	0.01	12.0
SO-5%	AR	50	10	0.01	2.0
SO-5%	Pyr	50	10	0.01	2.0
SO-5%	FS	50	10	0.005	12.0

S6.1.5. Additional visualizations and results

Figure S1 compares the three full-sequence smoothers under the cold-start protocol on a representative held-out trajectory. SDA takes the lead on this NS smoothing task: its vorticity panels reproduce the ground-truth small-scale eddies most faithfully, and its per-frame curves give the lowest NRMSE and spectrum relative error across the trajectory. ForcingDAS-FS is closely matched to SDA on both the field maps and the per-frame metrics, and its energy spectrum tracks the ground truth across wavenumbers; we acknowledge that SDA is the stronger smoother on this controlled PDE benchmark. EnKS-FS attenuates the small-scale modes (its predicted fields look diffused relative to the truth and its spectrum decays too quickly at high k), giving the largest per-frame errors of the three.

Table S6 extends the main-text NS table with the classical EnKF and EnKS baselines and reports the spectrum relative error in all four wavenumber bands (low-k, mid-k, high-k, and integrated all-k). FlowDAS preserves low-k and mid-k structure but accumulates large per-step error at high-k, where small-scale energy compounds beyond the context window. The classical EnKF and EnKS preserve total energy but their Gaussian analysis update blurs the large-scale modes, raising their low-k relative error. Across every wavenumber band and within every regime (Filtering, Fix-lag Smoothing, Full-sequence Smoothing), ForcingDAS variants attain the lowest NRMSE and the lowest spectrum relative error, with ForcingDAS-FS giving the strongest overall numbers.

It is important to note that the EnKF and EnKS results represent a strong classical baseline with access to the *exact* PDE dynamics. In contrast, ForcingDAS operates entirely with a learned prior and no explicit knowledge of the governing equations. Therefore, the reported EnKF and EnKS results serve as a reference for what is achievable by physics-based filters and smoothers with perfect model knowledge on this benchmark.

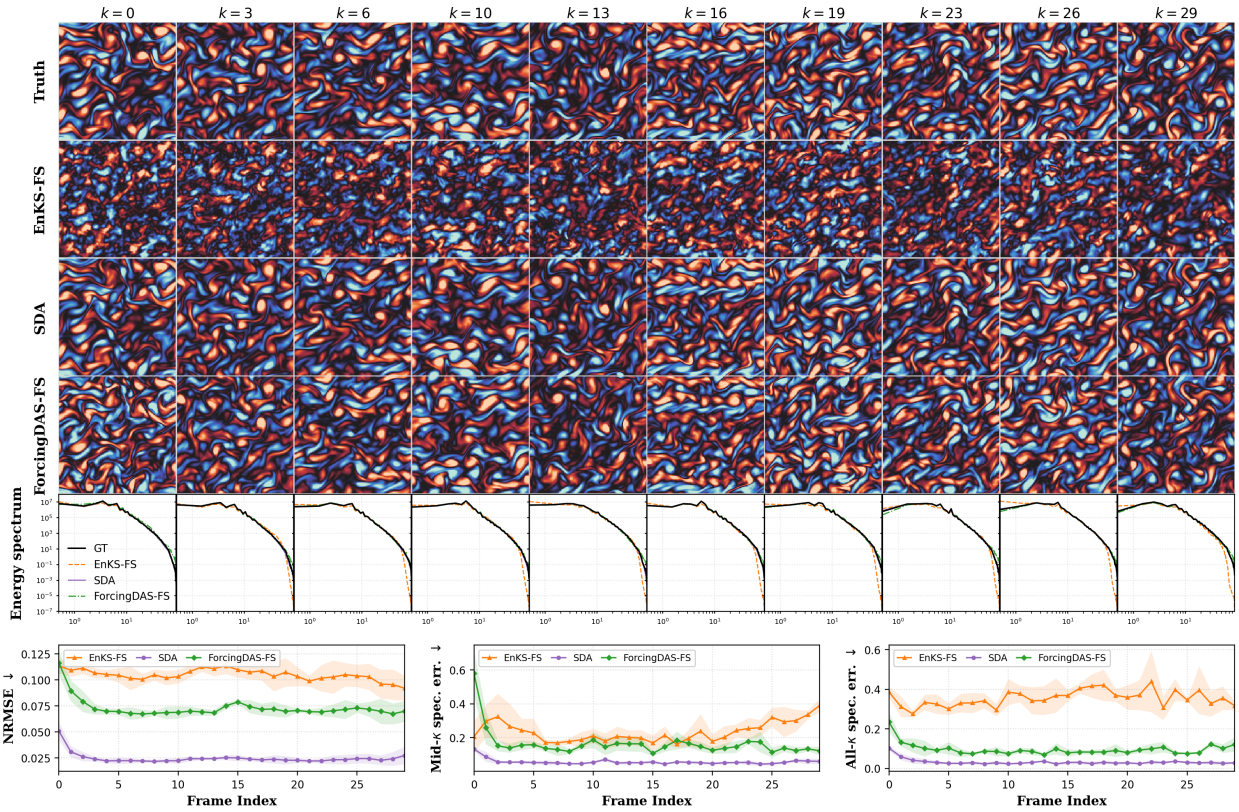

Figure S1. Full-sequence smoother comparison on a representative NS trajectory under SO-5% sparse-pixel observations with the no-context (cold-start) protocol. **Top four rows:** ground-truth vorticity field and per-method predictions for the classical Ensemble Kalman Smoother (EnKS-FS), the learned smoother SDA (Rozet & Louppe, 2023), and ForcingDAS-FS, at evenly-spaced frame indices $k \in \{0, 3, 6, 10, 13, 16, 19, 23, 26, 29\}$. **Fifth row:** per-frame radially-averaged kinetic-energy spectrum $E(k)$ on log-log axes, overlaying GT, EnKS-FS, SDA, and ForcingDAS-FS. **Bottom row:** per-frame NRMSE (↓), mid-k spectrum relative error (↓), and all-k spectrum relative error (↓). On this NS smoothing task, SDA takes the lead; ForcingDAS-FS closely matches SDA across the trajectory and across wavenumbers, while EnKS-FS attenuates the small-scale modes.

S6.2. SEVIR Radar Nowcasting

S6.2.1. 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 evaluate trajectories of length $K \in \{30, 49\}$ frames for the no-context and with-context protocols respectively, on 4 held-out test trajectories. 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. Per-threshold CSI numbers, the standard nowcasting skill metric, are reported in Table S8 (Appendix S6.2.5).

The raw VIL data are stored as 8-bit integer pixel intensities and are loaded as floating-point values pre-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 $[0, 1]$ reflectivity, 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 will give both conventions below.

S6.2.2. 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 mask retaining 10% resp. 20% of pixels) and SR2x, SR4x (bilinear down-sampling by 2 resp. 4).

Table S6. Navier–Stokes vorticity assimilation under SO-5% sparse-pixel observations, with 10 clean context frames. Mean NRMSE (↓) and the radially-averaged spectrum relative error (↓) at low-k ($k \in [0.5, 8)$), mid-k ($k \in [8, 32)$, the inertial range), high-k ($k \in [32, 64)$), and integrated all-k ($k \in [0.5, 64)$). Averaged over 4 held-out trajectories of length $K = 30$. The classical EnKF and EnKS-FixLag baselines use the same $K = 40$ with-context predictions as the other ctx=10 methods; the EnKS-FullSeq and SDA baselines are single offline passes over the $K = 30$ trajectory and match the cold-start protocol (no clean context), reported alongside the with-context columns since these offline smoothers do not distinguish a clean prefix from observations. SDA values are copied from Table S13; its low-k and high-k entries are not available ('–'). Cold-start results are in Appendix S7.1.

	Filtering			Fix-lag Smoothing		Full-sequence Smoothing		
Metric	FlowDAS	EnKF	ForcingDAS-AR	EnKS	ForcingDAS-Pyr	EnKS	SDA	ForcingDAS-FS
NRMSE	0.092	0.097	0.071	0.105	0.072	0.105	0.025	0.068
low-k	0.242	0.543	0.102	0.566	0.108	0.499	–	0.098
mid-k	0.278	0.412	0.150	0.234	0.159	0.235	0.058	0.139
high-k	1.792	0.780	0.275	0.517	0.279	0.501	–	0.248
all-k	1.031	0.612	0.207	0.417	0.213	0.401	0.097	0.188

Noise. Gaussian noise of standard deviation $\sigma_y^{\text{data}} = 0.05$ is added to the clean operator output, identically to NS.

Equivalent noise in physical units. Because the dataset's data space coincides with the $[0, 1]$ reflectivity convention ($b = 1$), Eq. (S8) 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):

$$\sigma_y^{\text{raw},\,[0,255]} \;=\; 0.05 \times 255 \;=\; 12.75. \tag{S12}$$

For reference, the lowest CSI threshold in the SEVIR evaluation suite is 16 on the same uint8 scale (Table S8), so the noise level is $\approx 80\%$ of the smallest nowcasting bin width and $\sim 6\%$ of the largest.

S6.2.3. 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.

S6.2.4. Sampling hyperparameters

Table S7. SEVIR sampling hyperparameters (DiT-B backbone, 100 DDIM steps, $\sigma_y^{\text{data}} = 0.05$).

Operator	Schedule	T (frames)	Context	γ	ζ
SO-10%	AR	30	0	0.010	1.0
SO-10%	Pyr	30	0	0.010	1.0
SO-10%	AR	49	6	0.005	1.0
SO-10%	Pyr	49	6	0.010	1.0
SO-20%	AR	30	0	0.005	0.7
SO-20%	Pyr	30	0	0.005	0.7
SO-20%	AR	49	6	0.007	0.7
SO-20%	Pyr	49	6	0.005	0.5
SR2x	AR	30	0	0.005	0.5
SR2x	Pyr	30	0	0.015	0.5
SR2x	AR	49	6	0.015	0.5
SR2x	Pyr	49	6	0.005	0.5
SR4x	AR	30	0	0.015	0.7
SR4x	Pyr	30	0	0.015	0.7
SR4x	AR	49	6	0.010	0.7
SR4x	Pyr	49	6	0.010	0.7

S6.2.5. PER-THRESHOLD CSI

Table S8 reports the per-threshold Critical Success Index (CSI) for ForcingDAS on SEVIR VIL across all four observation settings and both context protocols. Thresholds correspond to SEVIR VIL pixel intensity values; higher thresholds indicate heavier precipitation, which is increasingly rare and harder to reconstruct.

Table S8. SEVIR per-threshold CSI (↑) for FlowDAS and ForcingDAS under four observation settings, averaged over 4 held-out trajectories. AR = autoregressive (filtering, $u{=}K$); Pyr = pyramid (fixed-lag smoothing, $0{<}u{<}K$).

Obs	Ctx	Method	CSI-16	CSI-74	CSI-133	CSI-160	CSI-181	CSI-219	Mean
SO-10%	With	FlowDAS	0.764	0.785	0.605	0.464	0.412	0.307	0.556
		ForcingDAS-AR	0.880	0.858	0.664	0.533	0.491	0.361	0.631
		ForcingDAS-Pyr	0.881	0.856	0.665	0.533	0.493	0.354	0.630
		ForcingDAS-FS	0.888	0.868	0.682	0.559	0.522	0.379	0.650
	No	FlowDAS	0.533	0.558	0.472	0.378	0.365	0.280	0.431
		SDA	0.888	0.856	0.650	0.534	0.501	0.368	0.633
		ForcingDAS-AR	0.875	0.842	0.626	0.500	0.451	0.298	0.599
		ForcingDAS-Pyr	0.878	0.847	0.633	0.503	0.453	0.301	0.602
		ForcingDAS-FS	0.892	0.870	0.675	0.559	0.512	0.347	0.642
SO-20%	With	FlowDAS	0.799	0.850	0.705	0.547	0.452	0.263	0.603
		ForcingDAS-AR	0.905	0.891	0.732	0.622	0.591	0.455	0.699
		ForcingDAS-Pyr	0.900	0.888	0.730	0.618	0.588	0.461	0.697
		ForcingDAS-FS	0.912	0.897	0.739	0.636	0.611	0.491	0.714
	No	FlowDAS	0.647	0.713	0.631	0.521	0.470	0.333	0.552
		SDA	0.914	0.889	0.714	0.617	0.596	0.453	0.697
		ForcingDAS-AR	0.918	0.897	0.736	0.633	0.602	0.464	0.708
		ForcingDAS-Pyr	0.919	0.897	0.735	0.634	0.598	0.458	0.707
		ForcingDAS-FS	0.915	0.897	0.737	0.640	0.608	0.462	0.710
SR2x	With	FlowDAS	0.881	0.902	0.774	0.670	0.629	0.472	0.721
		ForcingDAS-AR	0.921	0.912	0.771	0.675	0.658	0.543	0.747
		ForcingDAS-Pyr	0.920	0.910	0.767	0.672	0.656	0.532	0.743
		ForcingDAS-FS	0.924	0.915	0.778	0.689	0.672	0.542	0.753
	No	FlowDAS	0.814	0.863	0.755	0.659	0.615	0.467	0.696
		SDA	0.917	0.896	0.734	0.648	0.637	0.489	0.720
		ForcingDAS-AR	0.932	0.914	0.772	0.691	0.665	0.525	0.750
		ForcingDAS-Pyr	0.935	0.919	0.777	0.695	0.676	0.535	0.756
		ForcingDAS-FS	0.936	0.923	0.788	0.706	0.684	0.531	0.761
SR4x	With	FlowDAS	0.814	0.808	0.613	0.477	0.443	0.330	0.581
		ForcingDAS-AR	0.794	0.848	0.649	0.522	0.485	0.347	0.607
		ForcingDAS-Pyr	0.814	0.849	0.647	0.517	0.476	0.336	0.606
		ForcingDAS-FS	0.882	0.865	0.677	0.540	0.506	0.345	0.636
	No	FlowDAS	0.533	0.586	0.504	0.382	0.334	0.244	0.431
		SDA	0.877	0.843	0.628	0.511	0.470	0.336	0.611
		ForcingDAS-AR	0.850	0.845	0.647	0.530	0.483	0.335	0.615
		ForcingDAS-Pyr	0.845	0.848	0.647	0.527	0.472	0.315	0.609
		ForcingDAS-FS	0.882	0.868	0.662	0.534	0.468	0.279	0.616

S6.2.6. ADDITIONAL VISUALIZATIONS AND RESULTS

The per-threshold Critical Success Index breakdown corresponding to Table 1 is reported in Table S8 (Appendix S6.2.5).

S6.3. ERA5 Global Atmospheric Reanalysis

S6.3.1. 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 on a 240×121 longitude–latitude grid with 6-hourly temporal resolution. The four channels are Z500, T850, U10, V10 (Appendix S6.3.2); we train on 1979–2015 and evaluate on 4 held-out length-30 trajectories from 2016. Appendix S1.2 surveys related learned-DA work on ERA5, and Appendix S8 describes our 3D-Var classical baseline.

The dataset stores values that are already z-score normalized per channel using climatological statistics from the training

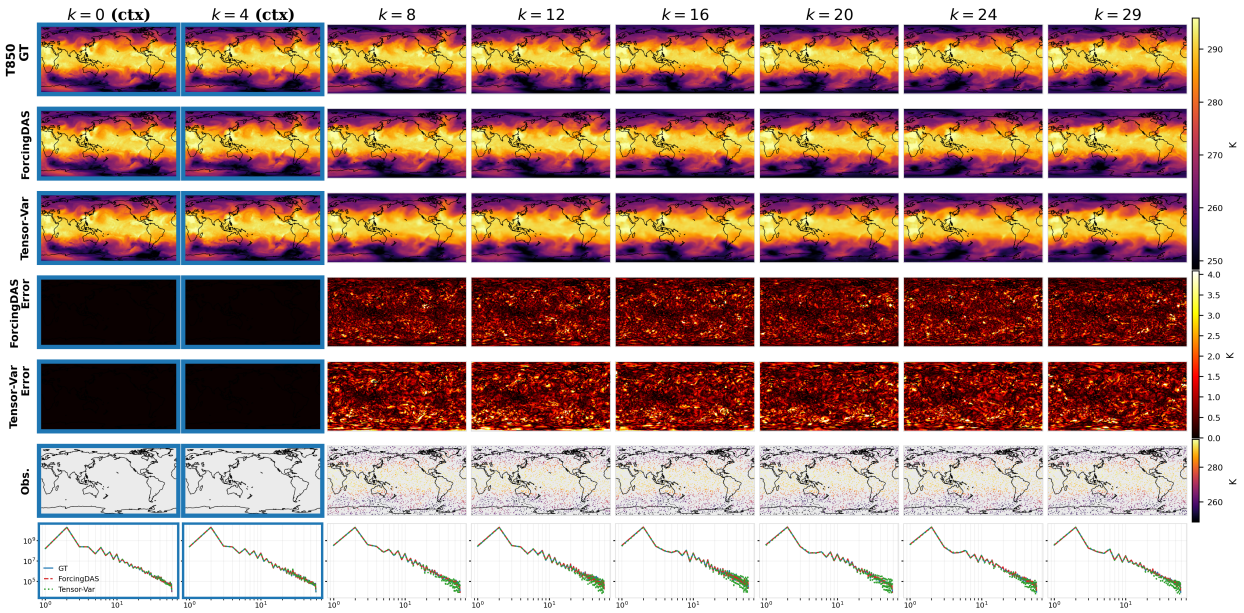

Figure S2. ERA5 SO-10% with-context assimilation, **T850** (temperature at 850 hPa) on the same held-out trajectory as Fig. 4. Row layout matches Fig. 4.

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 S9.

Table S9. 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.

Variable	μ_c	σ_c	Units
Z500	53 859.76	3,137.37	$\mathrm{m^2\,s^{-2}}$
T850	273.13	15.03	K
U10	−0.148	5.249	$\mathrm{m\,s^{-1}}$
V10	−0.224	4.410	$\mathrm{m\,s^{-1}}$

S6.3.2. Variable descriptions

Table S10 summarizes the four ERA5 variables considered for the multi-variable experiment (Option C in §4.3). Per-variable z-score normalization (subtract climatological mean, divide by climatological standard deviation, both computed from the training period) is applied before feeding data to the model, bringing all channels to approximately the same scale.

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. Importantly, the surface wind field is physically coupled to the upper-air geopotential via *geostrophic balance*: large-scale winds flow approximately parallel to geopotential contours, with speed proportional to the geopotential gradient. If ForcingDAS preserves this cross-variable coupling across filtering and smoothing regimes, it demonstrates that the framework maintains physical consistency beyond per-variable accuracy.

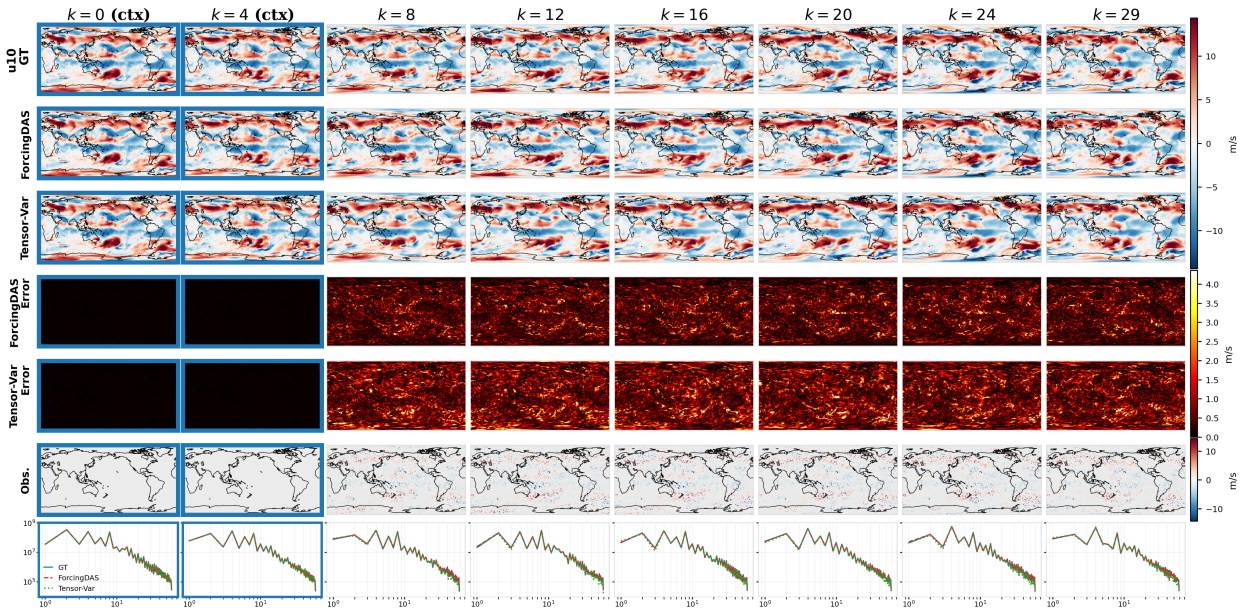

Figure S3. ERA5 SO-10% with-context assimilation, **U10** (zonal 10 m wind) on the same held-out trajectory as Fig. 4. Row layout matches Fig. 4.

Normalization. The four variables span three orders of magnitude in raw values (Table S10). We apply per-variable z-score standardization:

$$\tilde{x}_{t,c} = \frac{x_{t,c} - \mu_c}{\sigma_c}, \quad c \in \{\text{Z500, T850, U10, V10}\}, \tag{S13}$$

where μ_c and σ_c are the climatological mean and standard deviation of variable c, computed over the training period. 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).

S6.3.3. 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 observed ratio is 10% (SO-10%).

Noise. The data-space noise level is $\sigma_y^{\text{data}} = 0.05$. The noise is i.i.d. in space and across channels.

Equivalent per-channel noise in raw physical units. For ERA5, Eq. (S8) applies *per channel*, with $b = \sigma_c$ taken from Table S9: $\sigma_y^{\text{raw},c} = \sigma_c \cdot \sigma_y^{\text{data}}$. The operator (the same shared mask) is unchanged in raw space; the noise becomes anisotropic, with a different physical std per variable. Table S11 reports the resulting raw-space noise standard deviations. For example, $\sigma_y^{\text{data}} = 0.05$ corresponds to a noise of $\sim 157\,\text{m}^2\,\text{s}^{-2}$ on Z500 (a small fraction of typical synoptic anomalies, which are $\mathcal{O}(10^3)\,\text{m}^2\,\text{s}^{-2}$) and ~ 0.75 K on T850 (well below typical regional temperature variability).

S6.3.4. ERA5 evaluation metrics: NRMSE, ACC, and Bias

We report three per-variable evaluation metrics for ERA5 assimilation, all evaluated on the equiangular 240×120 longitude–latitude grid (after dropping the redundant 121st pole row) and all 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 $\hat{s}_{t,b,i,j}$ denote the assimilated value at frame t of trajectory b, longitude index $j \in \{1, \ldots, J\}$, latitude index $i \in \{1, \ldots, I\}$; let $s_{t,b,i,j}$ denote the ERA5 reference; and let $c_{t,b,i}$ denote the per-grid-point, per-time climatology (varying with day-of-year and hour-of-day). All quantities are in z-score-normalized data space (§S6.3.1).

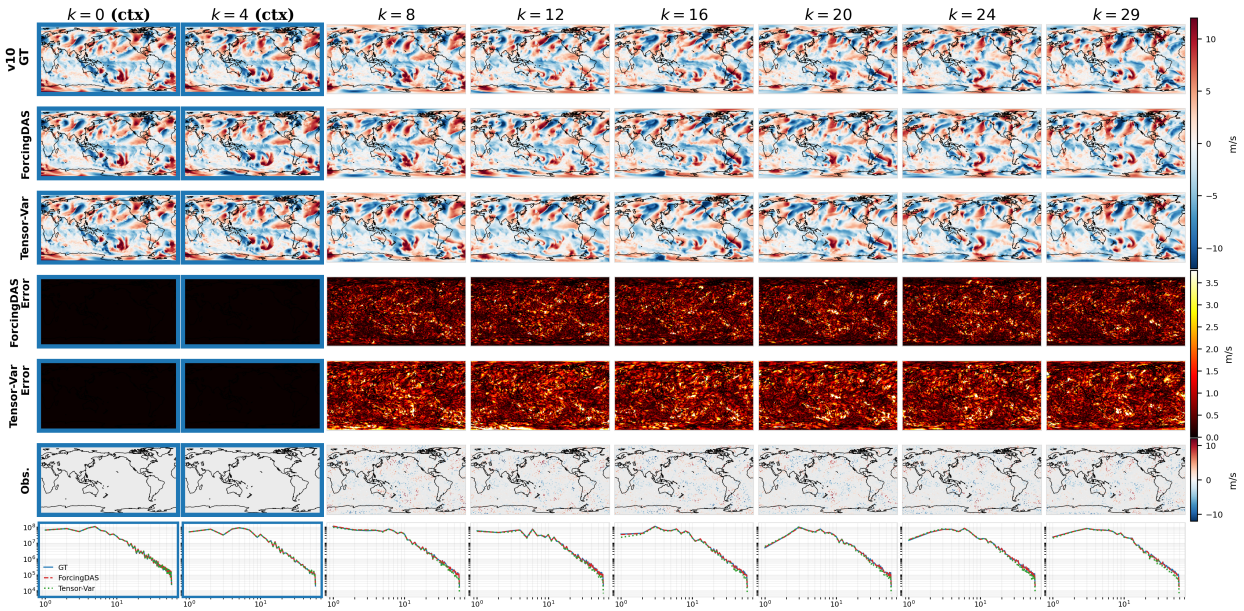

Figure S4. ERA5 SO-10% with-context assimilation, **V10** (meridional 10 m wind) on the same held-out trajectory as Fig. 4. Row layout matches Fig. 4.

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

$$w(i) = \frac{\cos\theta_i}{\frac{1}{I}\sum_{i'=1}^{I}\cos\theta_{i'}}, \tag{S14}$$

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 already 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, per-trajectory RMSE and then average:

$$\mathrm{NRMSE} = \frac{1}{BK}\sum_{b=1}^{B}\sum_{k=1}^{K}\sqrt{\frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} w(i)\left(\hat{s}_{k,b,i,j} - s_{k,b,i,j}\right)^2}, \tag{S15}$$

where $B = 4$ trajectories and $K = 30$ frames.

Latitude-weighted Bias (mean error). The Bias is the latitude-weighted mean signed error, averaged over all trajectories, frames, and spatial points:

$$\mathrm{Bias} = \frac{1}{B\cdot K\cdot I\cdot J}\sum_{b=1}^{B}\sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{j=1}^{J} w(i)\left(\hat{s}_{k,b,i,j} - s_{k,b,i,j}\right), \tag{S16}$$

reported per variable. Its sign carries information: positive entries indicate a systematic warm/high shift relative to ground truth, negative entries a cold/low shift; closer to zero is better. Bias is reported in the same z-score-normalized data space as NRMSE (§S6.3.1), so that magnitudes are directly comparable across the four variables; conversion back to raw physical units uses the per-channel scales of Table S9.

Anomaly Correlation Coefficient (ACC). We subtract the per-grid-point, per-time climatology to obtain anomalies $a_{k,b,i,j} = \hat{s}_{k,b,i,j} - c_{k,b,i}$ and $r_{k,b,i,j} = s_{k,b,i,j} - c_{k,b,i}$, and then compute the latitude-weighted Pearson correlation

Table S10. 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	$m^2\,s^{-2}$	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	$m\,s^{-1}$	-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	$m\,s^{-1}$	-20 to $+20$	Meridional (north–south) component of 10-meter wind. Together with U10, completes the surface wind vector.

Table S11. 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 S9.

σ_y^{data}	σ_y^{Z500} ($m^2\,s^{-2}$)	σ_y^{T850} (K)	σ_y^{U10} ($m\,s^{-1}$)	σ_y^{V10} ($m\,s^{-1}$)
0.05	156.87	0.752	0.262	0.221

between predicted and reference anomalies *jointly* over all trajectories, frames, and spatial points:

$$\text{ACC} = \frac{\sum_{k,b,i,j} w(i)\, a_{k,b,i,j}\, r_{k,b,i,j}}{\sqrt{\sum_{k,b,i,j} w(i)\, a_{k,b,i,j}^2}\,\sqrt{\sum_{k,b,i,j} w(i)\, r_{k,b,i,j}^2}}. \tag{S17}$$

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. Bias and ACC are complementary: a method can have $\text{ACC} \approx 1$ (correct anomaly pattern) yet a non-zero Bias (constant offset of the mean).

Climatology source. The climatology $c_{k,b,i}$ is the WeatherBench2 hourly ERA5 climatology (1990–2019 base period, 240×121 equiangular grid), indexed by the day-of-year and hour-of-day matching each frame's valid time. The climatology varies by latitude and calendar time but is constant across longitudes within each slice. The same fixed climatology is used for all methods compared in Table 2; in z-score data space the climatology is $c_{k,b,i}^{(\text{z})} = (c_{k,b,i}^{(\text{raw})} - \mu_c)/\sigma_c$ using the per-channel scales of Table S9.

S6.3.5. 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 across 4 GPUs (total batch size 16). Total training time is approximately 33 hours.

S6.3.6. Sampling hyperparameters

For ERA5 we use the same per-frame variance-reweighted DPS guidance as on NS and SEVIR, with (γ, ζ) tuned independently per setting. Table S12 lists the configurations used to produce the reported numbers at $\sigma_y^{\text{data}} = 0.05$ across the context $\times$ schedule grid ({ctx, no-ctx} $\times$ {AR, Pyr, FS}).

Table S12. ERA5 sampling hyperparameters across the context × schedule grid (DiT-B backbone, 100 DDIM steps, SO-10%, $\sigma_y^{\text{data}} = 0.05$). The setting IDs (S1–S4) correspond to the 4-cell AR/Pyr sweep (context × schedule) evaluated on 4 held-out trajectories; S5–S6 are the matching full-sequence (FS) configurations selected from a 5×6 ($\gamma \times \zeta$) Phase-1 sweep on the first held-out trajectory and committed at 4 trajectories.

Context	Schedule	ID	γ	ζ
6	Pyr	S1	0.005	1.000
6	AR	S2	0.005	1.000
0	Pyr	S3	0.005	1.000
0	AR	S4	0.005	1.000
6	FS	S5	0.001	4.000
0	FS	S6	0.005	8.000

S6.3.7. Additional visualizations and results

The figures below provide extended per-channel ERA5 panels (Z500, T850, U10, V10) for ForcingDAS-AR and ForcingDAS-Pyr against ground truth and the 3D-Var/FlowDAS baselines, together with per-variable zonal-wavenumber spectra in raw physical units.

Figures S2–S4 show the per-variable comparison panels for T850, U10, and V10 that complement Fig. 4 of the main text (same trajectory, same observation pattern, same row layout: ground truth, ForcingDAS prediction, TensorVar prediction, ForcingDAS error, TensorVar error, observations, and per-frame radially-averaged zonal-wavenumber spectrum).

S7. 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 (4 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.

S7.1. Navier–Stokes Vorticity

Table S13 reports cold-start NS results at SO-5%, including the SDA smoothing baseline that is not shown in the main text.

Table S13. Navier–Stokes vorticity, *cold-start* at SO-5% (no clean context, $K = 30$): mean NRMSE (↓) and the radially-averaged spectrum relative error (↓) at mid-k ($k \in [8, 32)$) and integrated all-k ($k \in [0.5, 64)$). Averaged over 4 held-out trajectories.

	Filtering		Fix-lag Smoothing	Smoothing	
Metric	FlowDAS	ForcingDAS-AR	ForcingDAS-Pyr	SDA	ForcingDAS-FS
NRMSE	0.114	0.083	0.081	0.025	0.073
mid-k	0.454	0.328	0.333	0.058	0.166
all-k	1.463	0.302	0.318	0.097	0.196

On NS smoothing at the SO-5% setting, SDA outperforms ForcingDAS-FS by a clear margin (NRMSE 0.025 vs. 0.073). NS is a Markovian, fully-observable benchmark: the vorticity field is the full state of the dynamical system, and the underlying PDE provides an exact per-step transition. This is the natural setting for SDA's bidirectional, frame-level score, which can use information from any direction without restriction. ForcingDAS uses causal temporal attention, which propagates information forward in time and is well-suited to filtering and fixed-lag smoothing; on the offline-smoothing endpoint, this directional bias is a small handicap. The picture reverses on the non-Markovian benchmarks below, where joint trajectory modeling becomes the dominant factor.

S7.2. Precipitation Nowcasting (SEVIR-VIL)

Table S14 reports cold-start SEVIR results.

Table S14. SEVIR VIL nowcasting, *cold-start* (no clean context, $K = 30$): NRMSE (↓), CSI-160 (↑), and CSI-mean (↑) over thresholds $\{16, 74, 133, 160, 181, 219\}$. Averaged over 4 held-out trajectories.

	SO-10%			SO-20%			SR2x			SR4x		
Method	NRMSE	CSI-160	CSI-m	NRMSE	CSI-160	CSI-m	NRMSE	CSI-160	CSI-m	NRMSE	CSI-160	CSI-m
Filtering												
FlowDAS	0.943	0.378	0.431	0.670	0.521	0.552	0.289	0.659	0.696	0.858	0.382	0.431
ForcingDAS-AR	0.348	0.500	0.599	0.240	0.633	0.708	0.175	0.691	0.750	0.332	0.530	0.615
Fix-lag Smoothing												
ForcingDAS-Pyr	0.332	0.503	0.602	0.238	0.634	0.707	0.171	0.695	0.756	0.331	0.527	0.609
Smoothing												
SDA	0.291	0.534	0.633	0.216	0.617	0.697	0.184	0.648	0.720	0.307	0.511	0.611
ForcingDAS-FS	0.288	0.559	0.642	0.222	0.640	0.710	0.153	0.706	0.761	0.291	0.534	0.616

In contrast to NS, SEVIR-VIL is a non-Markovian observed sequence: VIL is a vertically integrated radar product, a low-dimensional projection of a much higher-dimensional atmospheric state. Frame-to-frame learned dynamics (FlowDAS) degrade severely without a clean prefix to anchor the trajectory, with NRMSE roughly 2–3× higher than the with-context numbers in Table 1. ForcingDAS is stable across all four observation operators in all three

regimes. Comparing at apples-to-apples (both at cold-start), ForcingDAS-FS is competitive with SDA on NRMSE (FS wins on three of four operators and is essentially tied on SO-20%) and outperforms SDA on every operator on the precipitation-detection metrics CSI-160 and CSI-mean. The joint trajectory prior captures non-Markovian dependencies that SDA's per-frame score model cannot, and this advantage outweighs SDA's bidirectional information flow.

S7.3. Global Atmospheric State Estimation (ERA5)

Table S15 reports cold-start ERA5 results.

Table S15. ERA5 SO-10% assimilation, *cold-start* (no clean context, $K = 30$): latitude-weighted NRMSE (↓, normalized units), anomaly correlation coefficient (ACC, ↑), and latitude-weighted Bias (signed mean error; closer to 0 is better) per variable. Averaged over 4 held-out length-30 trajectories from 2016.

	Z500			T850			U10			V10		
Method	NRMSE	ACC	Bias	NRMSE	ACC	Bias	NRMSE	ACC	Bias	NRMSE	ACC	Bias
Filtering												
3D-Var	0.057	0.999	0.000	0.095	0.995	−0.001	0.393	0.927	0.009	0.438	0.901	0.001
ForcingDAS-AR	0.029	0.994	−0.001	0.061	0.968	−0.004	0.175	0.972	−0.002	0.213	0.971	−0.009
Fix-lag Smoothing												
ForcingDAS-Pyr	0.030	0.994	−0.001	0.062	0.967	−0.003	0.175	0.972	−0.001	0.214	0.971	−0.008
Smoothing												
TensorVar	0.049	0.984	−0.013	0.076	0.949	−0.012	0.220	0.954	0.000	0.275	0.950	−0.039
ForcingDAS-FS	0.024	0.996	0.000	0.055	0.973	−0.001	0.163	0.976	−0.001	0.202	0.974	−0.004

The ERA5 four-variable subset (Z500, T850, U10, V10) is also a non-Markovian observed sequence: the four channels are a small slice of the much higher-dimensional atmospheric state (37 vertical levels, dozens of variables) used by operational weather centers. ForcingDAS is competitive with or outperforms 3D-Var and TensorVar across all four variables and all three regimes, with cold-start NRMSE only modestly above the with-context numbers in Table 2 (e.g., ForcingDAS-FS Z500: 0.024 cold-start vs. 0.019 with-context). 3D-Var, by construction a per-frame spatial interpolator, achieves nearly identical NRMSE under cold-start and with-context, confirming that it carries no temporal propagation; ForcingDAS by contrast genuinely uses the assimilated trajectory.

S7.4. Discussion: Markovian vs. Non-Markovian Benchmarks

Taken together, the three benchmarks show a consistent pattern. On the Markovian, fully-observable NS benchmark, a specialized smoother (SDA) outperforms ForcingDAS-FS at offline smoothing. On the two non-Markovian benchmarks (SEVIR, ERA5), where the observed sequence is a low-dimensional projection of a higher-dimensional latent state, the ranking reverses: ForcingDAS matches or beats SDA and other strong baselines across regimes. This is consistent with the modeling claim in the introduction: the joint trajectory diffusion prior plus DiT-style temporal attention is the natural inductive bias for non-Markovian DA, where frame-level score models (SDA) and frame-to-frame transition models (FlowDAS) both miss the long-range dependencies that arise when the observed sequence is only a partial slice of the true state. The Markovian, fully-observable setting is where a baseline tailored for that single regime can still win, because the assumptions it relies on actually hold.

S8. Classical DA Baseline Implementation Details

We implement 3D-Var and 4D-Var baselines from scratch in PyTorch to ensure identical data pipelines, observation operators, and evaluation metrics as ForcingDAS. All operations are performed in z-score-normalized space.

S8.1. Shared Setup

Background state. In the no-context setting (matching ForcingDAS with `context_length` $= 0$), the background state is the climatological mean, which is $\boldsymbol{x}_b = \mathbf{0}$ in normalized space.

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 are generated by applying a shared binary spatial mask $\boldsymbol{m} \in \{0, 1\}^{H\times W}$ (identical across all C channels) and adding Gaussian noise:

$$\boldsymbol{y}_{k,c} = \boldsymbol{x}_{k,c} \odot \boldsymbol{m} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2\, \boldsymbol{I}), \tag{S18}$$

where $\sigma_y = 0.05$ in normalized space (matching the headline ForcingDAS setting in Table 2). The mask, noise realization, and random seed are shared with ForcingDAS for exact reproducibility.

Spatial grid. All data are cropped to spatial dimensions divisible by 8 (240×121 → 240×120), matching ForcingDAS's UNet downsampling requirement.

Optimization. 3D-Var uses L-BFGS in control space (see §S8 / 3D-Var subsection below); 4D-Var minimizes its cost function using Adam with learning rate 10^{-2}, up to 500 iterations with early stopping (patience 50, relative tolerance 10^{-7}).

S8.2. 3D-Var (Filtering)

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

$$J_{\text{3D}}(\boldsymbol{v}_k) = \frac{1}{2}\|\boldsymbol{v}_k\|^2 + \frac{1}{2\sigma_y^2}\sum_{c=1}^{C}\|\boldsymbol{y}_{k,c}[\boldsymbol{m}] - \boldsymbol{x}_{k,c}[\boldsymbol{m}]\|^2, \quad \boldsymbol{x}_k = \boldsymbol{x}_{b,k} + \sigma_b\, K_c * \boldsymbol{v}_{k,c}, \tag{S19}$$

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: $\boldsymbol{x}_{b,0} = \mathbf{0}$ and $\boldsymbol{x}_{b,k} = \boldsymbol{x}_{a,k-1}$ (persistence). This propagates information across time via the background field, allowing observations at time $t-1$ to inform the prior at time t. In the with-context setting, $\boldsymbol{x}_{b,k} = \boldsymbol{x}_{\text{gt},k}$ for $t < C$.

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 T frames are optimized jointly in a single batch (this is equivalent to per-frame optimization since the cost decomposes across time), followed by a second pass using the first-pass analysis as cycling background. Total wall-clock time is approximately 1 minute per trajectory on a single A800 GPU.

Interpretation. 3D-Var with Gaussian CVT produces a spatially smoothed analysis: the correlation kernel fills unobserved locations by interpolating from nearby observed pixels. This is effective for large-scale, spatially smooth fields (Z500, T850) where the dominant wavenumbers lie within the kernel bandwidth, yielding high ACC. However, the Gaussian smoothing inherently destroys structure at scales shorter than the correlation length, resulting in substantially higher NRMSE than learned methods. For sub-synoptic variables like surface winds (U10, V10), whose spatial scales approach or fall below the kernel width, 3D-Var's error grows to 2–7× that of ForcingDAS. Additionally, since 3D-Var performs no temporal propagation (only persistence cycling), it does not benefit from context frames beyond using the clean state as an initial background; comparing Table 2 (with-context) and Table S15 (cold-start), 3D-Var's NRMSE and ACC are nearly identical across the two protocols.

S8.3. 4D-Var (Smoothing)

Strong-constraint formulation. 4D-Var optimizes a single initial state $\boldsymbol{x}_0$ and propagates it forward through a deterministic forecast model $\mathcal{M}$:

$$J_{\text{4D}}(\boldsymbol{x}_0) = \frac{1}{2\sigma_b^2}\|\boldsymbol{x}_0\|^2 + \frac{1}{2\sigma_y^2}\sum_{k=0}^{K-1}\sum_{c=1}^{C}\|\boldsymbol{y}_{k,c}[\boldsymbol{m}] - \mathcal{M}^k(\boldsymbol{x}_0)_c[\boldsymbol{m}]\|^2, \tag{S20}$$

where $\mathcal{M}^k$ denotes k-fold composition of the single-step model ($\mathcal{M}^0 = \text{Id}$). Gradients $\nabla_{\boldsymbol{x}_0} J_{\text{4D}}$ are computed by backpropagating through the entire unrolled forecast chain via PyTorch autodiff, serving as an automatic adjoint model. After convergence, the full analysis trajectory is recovered by forward propagation: $\boldsymbol{x}_k^a = \mathcal{M}^k(\boldsymbol{x}_0^*)$.

Forecast model $\mathcal{M}$. $\mathcal{M}$ is a 2D UNet with residual prediction ($\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \text{UNet}(\boldsymbol{x}_k)$), designed to match ForcingDAS's spatial architecture:

- Base channels: 64, channel multipliers: $[1, 2, 4, 8]$ (matching ForcingDAS's 3D UNet).
- 2 residual blocks per level, GroupNorm, GELU activation.
- 29M parameters (ForcingDAS's 3D UNet has comparable spatial capacity plus temporal attention).

$\mathcal{M}$ is trained on the same ERA5 training split (1979–2015) with latitude-weighted MSE loss, AdamW optimizer (lr $= 3 \times 10^{-4}$, weight decay 10^{-4}), cosine learning rate schedule with 3-epoch warmup, mixed-precision training, and gradient clipping (max norm 1.0). Training uses distributed data-parallel across 4 GPUs with per-GPU batch size 16. During 4D-Var inference, the model parameters are frozen; only $\boldsymbol{x}_0$ is optimized.

Fairness considerations. The comparison between ForcingDAS and classical baselines is asymmetric by design:

- **Prior**: 3D-Var uses a Gaussian spatial correlation structure in $\boldsymbol{B}$ (CVT with per-variable isotropic kernels); ForcingDAS encodes rich multi-scale spatial priors through the learned diffusion model, capturing anisotropic, flow-dependent correlations that the fixed Gaussian kernel cannot represent.
- **Dynamics**: 3D-Var performs no temporal propagation (only persistence cycling); 4D-Var relies on a separately trained deterministic forecast model; ForcingDAS jointly learns dynamics and uncertainty through the diffusion process.
- **Architecture**: the forecast model $\mathcal{M}$ matches ForcingDAS's spatial backbone (same channel widths), ensuring comparable representational capacity for spatial features.

This asymmetry is the central experimental claim: a single learned generative model can outperform classical pipelines that require separate components for the prior, dynamics, and uncertainty.

S8.4. Ensemble Kalman Filter and Smoother Baselines for Navier–Stokes Experiment

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.

S8.4.1. Forward Model

The EnKF/EnKS forecast step integrates the same stochastic vorticity equation (S9) with the same forcing structure (S10) used to generate the truth data (see Appendix S6.1.1 for the full PDE specification). The forecast model uses a pseudo-spectral discretization on a 128×128 grid with a semi-implicit Euler–Maruyama time integrator at $\Delta t_{\text{sim}} = 2 \times 10^{-3}$: 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).

Model error. As described in Appendix S6.1.1, the truth data were generated at 256×256 resolution ($\Delta t = 10^{-4}$) and bilinearly downsampled to 128×128 (Chen et al., 2024d). 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.

S8.4.2. EnKF Configuration

Ensemble. We use $N_e = 100$ ensemble members, each initialized as a random snapshot drawn from all available trajectories in the dataset.

Forecast. Each ensemble member is integrated forward by $\Delta t_{\text{obs}} = 0.5$ time units (250 steps at $\Delta t_{\text{sim}} = 2 \times 10^{-3}$) using the stochastic NSE 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:

$$\boldsymbol{K} = \boldsymbol{P}^f \boldsymbol{H}^\top \left(\boldsymbol{H} \boldsymbol{P}^f \boldsymbol{H}^\top + \boldsymbol{R} \right)^{-1}, \tag{S21}$$

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.

S8.4.3. 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:

$$\boldsymbol{x}_{\text{aug}} = \begin{bmatrix} \boldsymbol{x}_{t-L} \\ \boldsymbol{x}_{t-L+1} \\ \vdots \\ \boldsymbol{x}_t \end{bmatrix} \in \mathbb{R}^{(L+1) \cdot n}, \tag{S22}$$

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, $L = 20$). The buffer holds the most recent 21 ensembles (current + 20 past). Each state estimate is refined by up to 20 future observations before being finalized and removed from the buffer. Evaluated on the same 40-frame trajectories as the EnKF (first 10 frames skipped).

EnKS-FS (full-sequence, $L = 29$). The buffer holds all ensembles from the entire trajectory (up to 30). Each state is refined by all subsequent observations with no early finalization. Evaluated on 30-frame trajectories (no frame skipping). Note that with $N_e = 100$ ensemble members and an augmented state of up to $30 \times 16{,}384 = 491{,}520$ dimensions, the sample covariance is severely rank-deficient, which limits the smoother's effectiveness and can degrade results compared to smaller windows.

S8.4.4. Observation Setup

Observations are generated identically to the ForcingDAS evaluation, 5% of the 128×128 grid points are randomly selected (819 points), with fixed observation locations across all time steps. Additive Gaussian noise with $\sigma_y = 0.05$ is applied, and the observation interval is $\Delta t_{\text{obs}} = 0.5$ time units between consecutive frames.

S8.4.5. Protocol

Four test trajectories are evaluated independently, with metrics averaged across trajectories. For the EnKF and EnKS-FL, the first 10 frames (indices 0–9) are skipped to match ForcingDAS, which uses these frames as context; assimilation proceeds for 39 cycles covering frames 10–49 (40 output frames including the initial analysis). For EnKS-FS, all 30 frames of the shorter test trajectories are used (29 assimilation cycles, no frame skipping).

S9. Limitations

This appendix expands the brief discussion in the main-text Limitations and Conclusion (§5).

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 S4). ForcingDAS is therefore strictly stronger than filtering but strictly weaker than a hypothetical bidirectional model that would allow direct forward-pass information flow in both directions. Removing this restriction, e.g., through bidirectional attention or a hybrid causal/bidirectional backbone, while preserving the single-model filtering–smoothing interpolation is an important open direction.

Compute at long horizons. Pyramid and full-sequence scheduling require jointly denoising multiple frames at every DDIM step, with memory and compute that scale with the active window size $W \approx K/u$. For very long trajectories, this cost can become prohibitive relative to the purely autoregressive regime, and blockwise or hierarchical schedulings may be needed.

Resolution and variables on ERA5. Our ERA5 study operates at 1.5° with four variables, sufficient to compare against the 3D-Var, FlowDAS, and TensorVar 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. ForcingDAS returns a single estimated trajectory per run; while the per-frame noise levels provide an *internal* uncertainty representation, we have not yet systematically evaluated the calibration of ensemble statistics obtained by running multiple seeds of the reverse process. A careful probabilistic evaluation on spread–error diagnostics is left to future work.