DAWIS: DATA ASSIMILATION WITH WINDOWED INVERSE SAMPLING VIA MULTITASK INTERPOLANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow- and diffusion-based generative models have recently emerged as flexible and highly efficient forecasting models for dynamical systems. When combined with inference-time guidance, they offer a promising route to high-dimensional non-Gaussian data assimilation (DA), the problem of combining forecasts with observations to estimate latent system states. Existing filters, however, condition on a fixed history and assimilate only the most recent observation, leaving them unable to revise past states when new observations arrive. Estimates then stay tethered to a history that later observations may contradict, and errors accumulate over the assimilation run. To this end, we introduce DAWIS, a unified DA method covering filtering, fixed-lag smoothing, and block smoothing within a single framework. DAWIS replaces the single flow time of a state-level prior with a multitask stochastic interpolant over a window of consecutive states, assigning a separate flow time to each. An assimilation cycle inverts the window to a vector of per-state turning points and regenerates it under observation guidance, with the turning points controlling how strongly each state is held fixed, revised, or generated from scratch. The same construction can also absorb the forecast into the assimilation cycle, removing the need for a separate forecasting model. Experiments on challenging nonlinear systems show that DAWIS improves on both filtering and smoothing baselines under sparse, noisy, and nonlinear observations.

1 INTRODUCTION

Many scientific and engineering problems, from weather forecasting and fluid mechanics to neuroscience and robotics, require estimating latent states from incomplete and noisy observations (Asch et al., 2016). Data assimilation (DA) combines prior knowledge of system dynamics with sparse, noisy observations $\boldsymbol{y}_{1:n} := (\boldsymbol{y}_1, \dots, \boldsymbol{y}_n)$ of a latent state sequence $\boldsymbol{x}_{0:n}$, where $\boldsymbol{y}_n$ relates to $\boldsymbol{x}_n$ through $p(\boldsymbol{y}_n|\boldsymbol{x}_n)$. DA comes in two complementary forms. *Filtering* tracks the *filtering distribution* $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$ online as observations arrive, whereas *smoothing* characterizes the *smoothing distribution* $p(\boldsymbol{x}_{0:N}|\boldsymbol{y}_{1:N})$ or its marginals $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:N})$ over a fixed window, using future observations to refine past estimates.

Both are particularly challenging in high-dimensional systems such as the atmosphere, where states evolve chaotically and observations are sparse, noisy, and heterogeneous. Existing methods for approximating these distributions can be broadly grouped by their formulation of inference and uncertainty representation. *Variational methods*, such as 4DVar, estimate the most probable trajectory over an assimilation window via optimization. They scale to very high-dimensional systems but traditionally require tangent-linear and adjoint models and primarily provide point estimates. *Ensemble Kalman methods*, including the EnKF and ensemble Kalman smoothers, propagate an ensemble to approximate filtering or smoothing distributions through their means and covariances. They are computationally attractive but rely on Gaussian or approximately Gaussian representations and often require careful tuning of localization and inflation (Calvello et al., 2024; Bannister, 2017). *Hybrid ensemble-variational methods*, such as En4DVar (Bonavita et al., 2012), combine variational optimization with flow-dependent ensemble covariances but retain limitations from finite ensembles and covariance-based uncertainty. Finally, *particle filters and smoothers* (Naesseth et al., 2019) can represent general distributions, but suffer from the curse of dimensionality (Bengtsson et al., 2008). These limitations motivate flexible, data-driven methods capable of representing complex non-Gaussian distributions while remaining computationally tractable in high-dimensional systems.

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

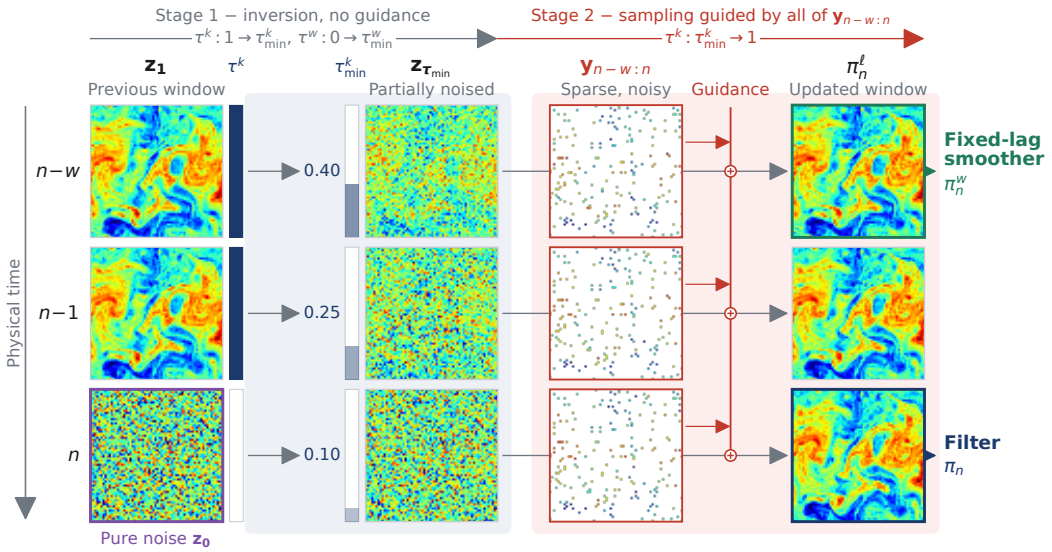


Figure 1: The DAWIS assimilation cycle, here visualized for DAWIS-Joint with a prior of $w+1 = 3$ states. (i) *Initialization:* The samples $\hat{\boldsymbol{x}}_{n-w:n-1}$ from the previous cycle start at flow time 1 and the new state $\boldsymbol{z_0} \sim \rho_0$. (ii) *Inversion:* the unconditional SDE moves each slot k to its turning point $\tau^k_{\min}$, partially noising the old states and running the new one forward from noise. (iii) *Guidance:* the guided SDE regenerates the window from $\boldsymbol{z}_{\boldsymbol{\tau}_{\min}}$, conditioned on observations $\boldsymbol{y}_{n-w:n}$.

Generative models provide and alternative route to high-dimensional sampling by representing flexible, non-Gaussian distributions and enable posterior sampling through inference-time conditioning. These ideas have been used to target filtering (Andrae et al., 2026; Chen et al., 2025) as well as smoothing (Rozet & Louppe, 2023; Jia et al., 2026). The choice of prior, however, largely dictates which of the two is available: methods that learn a stationary prior over states (Andrae et al., 2026) or short trajectories (Rozet & Louppe, 2023) and those that learn a forecast model $p(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n)$ (Chen et al., 2025; Savary et al., 2026) each commit to one regime. We generalize this construction from individual states to a *multitask trajectory prior*, assigning a separate flow time to each state in a window. This brings filtering, fixed-lag smoothing, and block smoothing into a single framework and removes the need for an external forecast model.

Contributions: We present DAWIS (**D**ata **A**ssimilation with **W**indowed **I**nverse **S**ampling), a unified framework for data assimilation built on flow-based generative models:

(i) **Windowed inversion.** We extend the inversion-based assimilation of Andrae et al. (2026) from a single state to whole windows. Each cycle inverts the previous estimate to a vector of per-state turning points $\boldsymbol{\tau}_{\min}$ and regenerates it under observation guidance, so new observations can revise past states instead of conditioning on a fixed history.

(ii) **A unified framework with one prior for several DA tasks.** Filtering, fixed-lag smoothing and block smoothing can all be done with the same trained model by changing the choice of $\boldsymbol{\tau}_{\min}$. A single sliding-window cycle gives both a filter and a lagged smoother.

(iii) **Forecasting with or without an external model.** DAWIS can incorporate an existing forecasting model or, with DAWIS-Joint, generate forecasts within the same generative inference process, eliminating the need for a separate model.

2 Preliminaries

2.1 Data Assimilation

Consider the state-space model

$$\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0), \qquad \boldsymbol{x}_n \sim p(\boldsymbol{x}_n \mid \boldsymbol{x}_{n-1}), \qquad \boldsymbol{y}_n \sim p(\boldsymbol{y}_n \mid \boldsymbol{x}_n), \qquad n = 1, \dots, N, \tag{1}$$

where $\boldsymbol{x}_n \in \mathbb{R}^d$ is the latent state and $\boldsymbol{y}_n$ a noisy, partial observation of it. The states form a Markov chain and the observations are conditionally independent given the states. The transition is written as a density for convenience, but is only ever accessed through simulation and may equally be a deterministic simulator or a pre-trained generative model.

The aim of DA is to infer the latent states $\boldsymbol{x}_{0:N}$ from the observations $\boldsymbol{y}_{1:N}$. The different DA tasks we consider are distinguished by which observations are available when estimating each state. Given observations up to the present, the target is the *filtering distribution* $\pi_n := p(\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n})$, computed sequentially as observations arrive. Allowing ℓ steps of look-ahead instead gives the *fixed-lag smoothing distribution* $\pi_n^\ell := p(\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n+\ell})$, which recovers the filter at $\ell = 0$ and admits more future information as ℓ grows, with diminishing returns once ℓ exceeds the forgetting time of the dynamics (Olsson et al., 2008). Using every observation gives the *joint smoothing distribution*, which by eq. (1) factorizes as

$$p(\boldsymbol{x}_{0:N} \mid \boldsymbol{y}_{1:N}) \propto p(\boldsymbol{x}_0) \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid \boldsymbol{x}_{n-1})\, p(\boldsymbol{y}_n \mid \boldsymbol{x}_n), \tag{2}$$

and whose marginals $p(\boldsymbol{x}_n \mid \boldsymbol{y}_{1:N})$ are the limit of the fixed-lag smoother as ℓ grows. Computing these targets exactly is intractable for the high-dimensional, nonlinear systems we consider. Our aim is to develop a practical approximation that cover all of them in a single unified framework.

2.2 Multitask Stochastic Interpolants

Our proposed DA method leverages recent advances in generative modeling. In the static setting (i.e., ignoring time evolution), flow-based models such as stochastic interpolants (Albergo et al., 2023) enable sampling from a target distribution ρ_1 by learning a transport map from a simple latent distribution ρ_0. In their standard form, these constructions transport the entire sample along a single time axis. However, when ρ_1 is a distribution over a trajectory of states $\boldsymbol{x}_{n:n+w}$, we want access not only to the joint distribution but also to its conditionals. This can be achieved by assigning a separate time to each state in the window, allowing the states to be noised and denoised at different rates. In this work, we adopt the framework of multitask interpolants (Negrel et al., 2025), though closely related formulations appear in the diffusion literature (Chen et al., 2024).

We define a *multitask interpolant* as the stochastic process

$$\boldsymbol{z}_{\boldsymbol{\tau}} = \alpha_{\boldsymbol{\tau}} \odot \boldsymbol{z}_{\mathbf{0}} + \beta_{\boldsymbol{\tau}} \odot \boldsymbol{z}_{\mathbf{1}}, \qquad \boldsymbol{\tau} \in [0,1]^D, \tag{3}$$

where $\boldsymbol{z}_{\mathbf{0}} \sim \rho_{\mathbf{0}}$ and $\boldsymbol{z}_{\mathbf{1}} \sim \rho_{\mathbf{1}}$, and each entry of $\boldsymbol{\tau}$ specifies the *flow time* of the corresponding component. The scalar curves $\alpha, \beta \in C^2([0,1])$ satisfy $\alpha(0) = \beta(1) = 1$ and $\alpha(1) = \beta(0) = 0$ and all evaluations and multiplications are understood elementwise.

In order to use the multitask interpolant for sampling, we must specify how the component-wise flow times evolve during generation. We therefore introduce a path $\boldsymbol{\tau}_t \in C^2([0,1]; [0,1]^D)$ and consider the time-indexed interpolant $\boldsymbol{z}_{\boldsymbol{\tau}_t}$. Different choices of $\boldsymbol{\tau}_t$ give rise to different sampling procedures. For example, $\boldsymbol{\tau}_t = t\mathbf{1}$ recovers the standard stochastic interpolant, while fixing $\tau_t^k \equiv 1$ for $k \in \mathcal{K}$ and setting the corresponding data components to $\boldsymbol{z}_1^k = \boldsymbol{x}_k$ yields samples from $p(\boldsymbol{x} \mid \boldsymbol{x}_{\mathcal{K}})$. Likewise, choosing a decreasing path ($\dot{\boldsymbol{\tau}}_t \leq 0$) reverses the transport, mapping data toward the latent space.

Let $\rho_{\boldsymbol{\tau}_t} := \mathrm{Law}(\boldsymbol{z}_{\boldsymbol{\tau}_t})$ denote the distribution of $\boldsymbol{z}_{\boldsymbol{\tau}_t}$ with density $p_{\boldsymbol{\tau}_t}$, and define its *score* $\boldsymbol{s}_{\boldsymbol{\tau}}(t, \boldsymbol{z}) := \nabla \log p_{\boldsymbol{\tau}_t}(\boldsymbol{z})$ and *drift* $\boldsymbol{b}_{\boldsymbol{\tau}}(t, \boldsymbol{z}) := \mathbb{E}[\dot{\boldsymbol{z}}_{\boldsymbol{\tau}_t} | \boldsymbol{z}_{\boldsymbol{\tau}_t} = \boldsymbol{z}]$. Then Negrel et al. (2025) show that for any non-negative $\boldsymbol{\varepsilon}_{\boldsymbol{\tau}} \in C^0([0,1]; \mathbb{R}_+^D)$, the SDE

$$\mathrm{d}\boldsymbol{z}_{\boldsymbol{\tau}_t} = (\boldsymbol{b}_{\boldsymbol{\tau}}(t, \boldsymbol{z}_{\boldsymbol{\tau}_t}) + \boldsymbol{\varepsilon}_{\boldsymbol{\tau}_t} \odot \boldsymbol{s}_{\boldsymbol{\tau}}(t, \boldsymbol{z}_{\boldsymbol{\tau}_t}))\mathrm{d}t + \sqrt{2\boldsymbol{\varepsilon}_{\boldsymbol{\tau}_t}} \odot \mathrm{d}W_t, \quad \boldsymbol{z}_{\boldsymbol{\tau}_0} \sim \rho_{\boldsymbol{\tau}_0}, \quad t \in [0,1], \tag{4a}$$

where W_t is Brownian motion, has the same marginal laws $\rho_{\boldsymbol{\tau}_t}$ as the interpolant $\boldsymbol{z}_{\boldsymbol{\tau}_t}$. Sampling from eq. (4) requires access to the drift and score. Rather than learning these quantities separately for each path $\boldsymbol{\tau}_t$, we learn the path-independent conditional expectations

$$\boldsymbol{\eta}_{\mathbf{0}}(\boldsymbol{\tau}, \boldsymbol{z}) := \mathbb{E}[\boldsymbol{z}_{\mathbf{0}} \mid \boldsymbol{z}_{\boldsymbol{\tau}} = \boldsymbol{z}], \qquad \boldsymbol{\eta}_{\mathbf{1}}(\boldsymbol{\tau}, \boldsymbol{z}) := \mathbb{E}[\boldsymbol{z}_{\mathbf{1}} \mid \boldsymbol{z}_{\boldsymbol{\tau}} = \boldsymbol{z}], \tag{5}$$

from which the drift and score can be recovered for any chosen path. The resulting sampling procedure therefore supports different paths without retraining. Details of the training objectives and the corresponding drift and score expressions are given in Appendix B.1.

2.3 Conditional Generation via Guidance

Given an observation $\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{x})$ and the SDE in eq. (4), we can sample from the posterior $\rho_{\mathbf{1}}^{\boldsymbol{y}}(\boldsymbol{x}) \propto p(\boldsymbol{y}|\boldsymbol{x})\rho_{\mathbf{1}}(\boldsymbol{x})$ by solving the following SDE with *guidance*:

$$\mathrm{d}\boldsymbol{z}_{\boldsymbol{\tau}_t} = (\boldsymbol{b}_{\boldsymbol{\tau}}(t, \boldsymbol{z}_{\boldsymbol{\tau}_t}; \boldsymbol{y}) + \boldsymbol{\varepsilon}_{\boldsymbol{\tau}_t} \odot \boldsymbol{s}_{\boldsymbol{\tau}}(t, \boldsymbol{z}_{\boldsymbol{\tau}_t}; \boldsymbol{y}))\mathrm{d}t + \sqrt{2\boldsymbol{\varepsilon}_{\boldsymbol{\tau}_t}} \odot \mathrm{d}W_t, \quad \boldsymbol{z}_{\boldsymbol{\tau}_0} \sim \rho_{\boldsymbol{\tau}_0}, \quad t \in [0,1], \tag{6a}$$

where guided score and drift are given by

$$\boldsymbol{s}_{\boldsymbol{\tau}}(t, \boldsymbol{z}_{\boldsymbol{\tau}_t}; \boldsymbol{y}) = \boldsymbol{s}_{\boldsymbol{\tau}}(t, \boldsymbol{z}_{\boldsymbol{\tau}_t}) + \nabla_{\boldsymbol{z}_{\boldsymbol{\tau}_t}} \log p(\boldsymbol{y}|\boldsymbol{z}_{\boldsymbol{\tau}_t}), \tag{7a}$$

$$\boldsymbol{b}_{\boldsymbol{\tau}}(t, \boldsymbol{z}_{\boldsymbol{\tau}_t}; \boldsymbol{y}) = \boldsymbol{b}_{\boldsymbol{\tau}}(t, \boldsymbol{z}_{\boldsymbol{\tau}_t}) + \boldsymbol{\lambda}_{\boldsymbol{\tau}_t} \odot \nabla_{\boldsymbol{z}_{\boldsymbol{\tau}_t}} \log p(\boldsymbol{y}|\boldsymbol{z}_{\boldsymbol{\tau}_t}), \tag{7b}$$

for $\boldsymbol{\lambda}_{\boldsymbol{\tau}_t}$ defined as in Appendix B.2. This also implies that the SDE eq. (6) share the same marginal laws $\rho_{\boldsymbol{\tau}_t}^{\boldsymbol{y}}$ as the interpolant eq. (3) where the endpoint $\boldsymbol{z}_{\mathbf{1}}$ is a sample from the posterior $\rho_{\mathbf{1}}^{\boldsymbol{y}}$.

In general, the guidance term is intractable as we usually only have access to $p(\boldsymbol{y} \mid \boldsymbol{z}_{\mathbf{1}})$ but not $p(\boldsymbol{y} \mid \boldsymbol{z}_{\boldsymbol{\tau}_t}) = \mathbb{E}_{\boldsymbol{z}_{\mathbf{1}}|\boldsymbol{z}_{\boldsymbol{\tau}_t}}[p(\boldsymbol{y} \mid \boldsymbol{z}_{\mathbf{1}})]$. Various approaches exist to approximate this term, including diffusion posterior sampling (DPS) Chung et al. (2023), moment matching posterior sampling (MMPS; Rozet et al., 2024), and recent asymptotically exact methods such as Meta Flow Maps (Potaptchik et al., 2026). We use MMPS for its simplicity, but note that any gradient-based guidance method can be used within the DAWIS framework. See Appendix B.2 for details and Daras et al. (2024) for a survey on guidance methods.

3 Method

DAWIS is a unified approach to filtering and smoothing in high-dimensional non-Gaussian DA, built on a single learned prior over windows of states $p(\boldsymbol{x}_{n:n+w})$. The prior is parameterized as a multitask interpolant and is learned from trajectory data. We assume a single flow time for each state within the window, giving a vector $\boldsymbol{\tau} = (\tau^0, \dots, \tau^w)$ of $w+1$ flow times. This enables generating a state from noise, holding it fixed, or anything in between, independently for each state in the window. This flexibility allows us to simultaneously: (i) retain information from previous assimilation cycles for efficiency and temporal consistency, by *partial* inversion and regeneration, and (ii) give the method sufficient flexibility to update old state variables as new data arrives.

DAWIS is an ensemble method that generates J independent samples $\{\hat{\boldsymbol{x}}_{:}^{j}\}_{j=1}^{J}$ to approximate the filtering or smoothing distribution as an empirical distribution. We describe the methodology for generating a single sample $\hat{\boldsymbol{x}}$. Each DAWIS assimilation cycle consists of three stages: initialization, inversion, and guidance. We build up the methodology by exemplifying these stages for filtering and fixed-lag smoothing, guided forecast models, and block smoothing, respectively, in the three subsections below. Algorithm 1 presents the general method and table 1 the design choices that recover the different instances.

3.1 Filtering and Fixed-lag Smoothing

Consider first the filtering problem of sampling sequentially from π_n. One option is to guide a generative forecast model trained to simulate from $p(\boldsymbol{x}_n \mid \boldsymbol{x}_{n-1})$ to target the conditional $p(\boldsymbol{x}_n \mid \boldsymbol{y}_n, \boldsymbol{x}_{n-1})$ (see further section 3.2) (Chen et al., 2025; Savary et al., 2026). However, this requires conditioning the simulation on a current sample $\boldsymbol{x}_{n-1}$, which makes the assimilation locked to a particular history. A remedy is to regenerate a longer window, simulating $p(\boldsymbol{x}_{n-w+1:n} \mid \boldsymbol{y}_{n-w+1:n}, \boldsymbol{x}_{n-w})$ (Shysheya et al., 2024). However, this simulation is still based on a hard conditioning on a historical state, just further back in time, and furthermore requires regenerating the whole window from scratch.

An alternative is to use the *inversion strategy* introduced in the DAISI method by Andrae et al. (2026). They start by producing a forecast $\boldsymbol{x}_n^{\mathrm{f}}$, using an off-the-shelf forecasting model, conditionally on the sample $\boldsymbol{x}_{n-1}$. The forecast state is then inverted to a latent representation $\boldsymbol{z}_{\tau_{\min}}$ by simulating the SDE in equation 4 from $\tau_0 = 1$ to $\tau_{\min}$ (with the prior restricted to a single state). The inversion is followed by a guided "regeneration" of $\hat{\boldsymbol{x}}_n$ using the guided SDE in equation 6, running from $\tau_{\min}$ to $\tau = 1$, while conditioning on $\boldsymbol{y}_n$.

DAWIS generalizes this inversion strategy to windowed sampling, where the inversion of each state within the window can be controlled independently. Let $\mathcal{W} = \{n-w, \dots, n\}$ denote the window at

Algorithm 1 DAWIS – one assimilation cycle

Require: current states $\boldsymbol{x}_{\mathcal{W}}$, observations $\boldsymbol{y}_{\mathcal{W}}$, start times $\boldsymbol{\tau}_0$, turning points $\boldsymbol{\tau}_{\min}$

1: **Initialization:** Set $\boldsymbol{z}_{\boldsymbol{\tau}_0} \leftarrow \alpha_{\boldsymbol{\tau}_0} \odot \boldsymbol{z_0} + \beta_{\boldsymbol{\tau}_0} \odot \boldsymbol{x}_{\mathcal{W}}$ for $\boldsymbol{z_0} \sim \rho_{\boldsymbol{0}}$

2: **Inversion:** $\boldsymbol{z}_{\boldsymbol{\tau}_{\min}} \leftarrow \textsc{Integrate}\big(\boldsymbol{z}_{\boldsymbol{\tau}_0};\ \boldsymbol{\tau}_0 \to \boldsymbol{\tau}_{\min}\big)$ ▷ unconditional

3: **Guidance:** $\boldsymbol{z_1} \leftarrow \textsc{Integrate}\big(\boldsymbol{z}_{\boldsymbol{\tau}_{\min}};\ \boldsymbol{\tau}_{\min} \to \boldsymbol{1},\ \boldsymbol{y}_{\mathcal{W}}\big)$ ▷ conditional on $\boldsymbol{y}_{\mathcal{W}}$

4: **return** $\hat{\boldsymbol{x}}_{\mathcal{W}} \leftarrow \boldsymbol{z_1}$

assimilation cycle n. From the previous cycle, each ensemble member $\hat{\boldsymbol{x}}_{n-w-1:n-1}$ is an approximate draw from $p(\boldsymbol{x}_{n-w-1:n-1} \mid \boldsymbol{y}_{1:n-1})$. We **initialize** cycle n by shifting the sample by one time step and augmenting it with a forecast $\boldsymbol{x}_n^{\mathrm{f}}$. As in DAISI, this can be obtained using an off-the-shelf forecast model, but an alternative is to use the DAWIS prior itself; see section 3.2 below. This gives the initial state $\boldsymbol{z_1} = [\hat{\boldsymbol{x}}_{n-w:n-1}, \boldsymbol{x}_n^{\mathrm{f}}]$, at flow times $\boldsymbol{\tau} = \boldsymbol{1}$. This follows from the fact that, with this initialization, all states within the window "live in the ambient state space", corresponding to $\tau = 1$.

Next, analogously to DAISI, the state $\boldsymbol{z_1}$ is **inverted** to a latent representation. However, for the *windowed inversion*, each slot k within the window is assigned its own *turning point* $\tau_{\min}^k \in [0, 1]$, collected in $\boldsymbol{\tau}_{\min} = (\tau_{\min}^0, \ldots, \tau_{\min}^w)$. The unconditional SDE in eq. (4) is integrated backwards from $\boldsymbol{\tau} = \boldsymbol{1}$ to $\boldsymbol{\tau}_{\min}$. The resulting latents $\boldsymbol{z}_{\boldsymbol{\tau}_{\min}}$ encode the information in the current sample in an intermediate latent space. Finally, we perform **guided** sampling given $\boldsymbol{y}_{\mathcal{W}}$ by integrating the guided SDE in eq. (6) from $\boldsymbol{\tau}_{\min}$ back to $\boldsymbol{\tau} = \boldsymbol{1}$. This produces an assimilated sample $\hat{\boldsymbol{x}}_{\mathcal{W}}$.

This formulation combines the inversion strategy with windowed sampling. Importantly, by choosing a different turning point for each state, we have individual control over how far each state is inverted. Setting $\tau_{\min}^k = 1$ freezes slot k, while smaller values invert the slot further, allowing for more change.

The same procedure does not only produce filtering estimates, but also fixed-lag smoothing estimates. Because the window spans $w+1$ physical times, slot w of the window holds the current state and gives an approximate filter $\boldsymbol{x}_n \sim \pi_n$. Slot 0, on the other hand, holds the state w steps back and gives a fixed-lag smoother $\boldsymbol{x}_{n-w} \sim \pi_{n-w}^w$, which has been constrained by the w observations that arrived since. We refer to these as *DAWIS Filter* and *DAWIS Lagged Smoother*. The remaining slots slide into cycle $n+1$, so each state remains in the window for $w+1$ cycles and is revised by every observation that enters behind it.

Table 1: Entries marked $*$ are irrelevant for $\beta_0 = 0$ as $\boldsymbol{\tau}_{\min} = \boldsymbol{\tau}_0$ means no inversion.

Method	$\boldsymbol{x}_{\mathcal{W}}$	$\boldsymbol{\tau}_0$	$\boldsymbol{\tau}_{\min}$	$\boldsymbol{y}_{\mathcal{W}}$
DAWIS Filter / Lagged Sm.	$[\hat{\boldsymbol{x}}_{n-w:n-1}, \boldsymbol{x}_n^{\mathrm{f}}]$	$(1, \ldots, 1)$	$\boldsymbol{\tau}_{\min}$	$\boldsymbol{y}_{n-w:n}$
DAWIS-Joint	$[\hat{\boldsymbol{x}}_{n-w:n-1}, *]$	$(1, \ldots, 1, 0)$	$\boldsymbol{\tau}_{\min}$	$\boldsymbol{y}_{n-w:n}$
DAWIS Block Smoother	$[\boldsymbol{x}_{n:n+w}]$	$(1, \tau_{\min}, \ldots, \tau_{\min}, 1)$	$\boldsymbol{\tau}_0$	$\boldsymbol{y}_{\mathcal{B}}$
Block Gibbs Smoother	$[\boldsymbol{x}_{n:n+w}]$	$(1, 0, \ldots, 0, 1)$	$\boldsymbol{\tau}_0$	$\boldsymbol{y}_{\mathcal{B}}$
Joint AR $w\|1$	$[\hat{\boldsymbol{x}}_{n-w}, *, \ldots, *]$	$(1, 0, \ldots, 0)$	$\boldsymbol{\tau}_0$	$\boldsymbol{y}_{n-w+1:n}$
Joint AR $1\|w$	$[\hat{\boldsymbol{x}}_{n-w:n-1}, *]$	$(1, \ldots, 1, 0)$	$\boldsymbol{\tau}_0$	$\boldsymbol{y}_n$

3.2 Guided Forecast models

The procedure above assumes access to a forecast $\boldsymbol{x}_n^{\mathrm{f}}$ from an external forecasting model $\mathcal{F}$, which in many applications may not be available. Since the multitask interpolant can sample from the conditional $p(\boldsymbol{x}_n \mid \boldsymbol{x}_{n-1})$, we could use this as a drop-in replacement for $\mathcal{F}$. However, this would require a separate sampling step, increasing the cost of a cycle. Instead, inspired by guided forecast models, we propose to initialize the final slot w from noise at flow time $\tau^w = 0$, skipping the forecast stage entirely. During the inversion stage, this slot is now run *forward* to $\tau_{\min}^w$, yielding a $\boldsymbol{z}_{\boldsymbol{\tau}_{\min}}$ similarly to before. The guidance step remains the same, generating the posterior directly without an extra step. We name this variant *DAWIS-Joint* and visualize it in fig. 1.

More generally, the SDE eq. (4) can be initialized from any $\rho_{\boldsymbol{\tau}_0}$, allowing us to generate one or more states from pure noise. In particular, the windowed guided forecast model $p(\boldsymbol{x}_{n-w+1:n} \mid \boldsymbol{y}_{n-w+1:n}, \boldsymbol{x}_{n-w})$ is itself an instance of DAWIS, obtained by freezing slot 0 at $\boldsymbol{x}_{n-w}$, initializing

$\boldsymbol{x}_{n-w+1:n}$ from noise, and skipping the inversion stage. As for the DAWIS Lagged Smoother, slot 1 has been constrained by $\boldsymbol{y}_{n-w+1:n}$, giving a fixed-lag smoother with lag $w-1$, $\boldsymbol{x}_{n-w+1} \sim \pi^{w-1}_{n-w+1}$.

3.3 Block Smoothing

The DAWIS Lagged Smoother performs sequential assimilation by sliding a window forward in time. Although it can utilize future observations, the fixed lag limits the possibility to propagate information from observations outside the window.

Rather than retraining a prior over increasingly long windows, which can become infeasible for high-dimensional systems and requires a separate prior for each trajectory length, we introduce a block smoother that iteratively refines the trajectory while reusing the same prior.

For a window $\boldsymbol{x}_{n:n+w}$ with interior $\mathcal{B} = \{n+1, \dots, n+w-1\}$ and boundary $\partial\mathcal{B} = \{n, n+w\}$, the Markov structure of eq. (1) implies that

$$\pi_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}) := p(\boldsymbol{x}_{\mathcal{B}} \mid \boldsymbol{x}_{0:N\setminus\mathcal{B}}, \boldsymbol{y}_{1:N}) = p(\boldsymbol{x}_{\mathcal{B}} \mid \boldsymbol{x}_{\partial\mathcal{B}}, \boldsymbol{y}_{\mathcal{B}}) \propto p(\boldsymbol{x}_{\mathcal{B}} \mid \boldsymbol{x}_{\partial\mathcal{B}})\, p(\boldsymbol{y}_{\mathcal{B}} \mid \boldsymbol{x}_{\mathcal{B}}). \tag{8}$$

Repeatedly drawing from eq. (8) for different blocks is a blocked Gibbs sampler for the joint smoothing distribution eq. (2) (Carter & Kohn, 1994; Singh et al., 2017). Each draw is available from the multitask interpolant by holding the boundary slots at $\boldsymbol{\tau}^{\partial\mathcal{B}} = \mathbf{1}$ and generating the interior from pure noise under guidance from $\boldsymbol{y}_{\mathcal{B}}$. To ensure every state is updated, we use overlapping blocks, alternating forward and backward sweeps. Given exact guidance, this *Block Gibbs Smoother* converges to eq. (2) from any initial trajectory (Carter & Kohn, 1994).

In practice, the guidance method will incur a bias, limiting the ability to converge to the true posterior despite many iterations. Instead, we use the block smoother to refine an already assimilated trajectory, using the alternating sweeps to propagate information from observations further away. However, since the interior is resampled from pure noise, much of the information is thrown away at every iteration, limiting its effectiveness.

To remedy this, we propose the *DAWIS Block Smoother*, an extension of the DAWIS update to the block smoothing setting. Instead of discarding the previous sample, we initialize the interior $\boldsymbol{z}^{\mathcal{B}}_{\boldsymbol{\tau}_{\min}}$ from the interpolant

$$\boldsymbol{z}^{\mathcal{B}}_{\boldsymbol{\tau}_{\min}} = \alpha_{\tau_{\min}}\, \boldsymbol{z}_{\mathbf{0}} + \beta_{\tau_{\min}}\, \boldsymbol{x}_{\mathcal{B}}, \qquad \boldsymbol{z}_{\mathbf{0}} \sim \rho_{\mathbf{0}}. \tag{9}$$

preserving part of the information from the previous iteration. We then proceed the same way as before, running the guidance step conditioned on observations $\boldsymbol{y}_{\mathcal{B}}$.

For this to be a valid replacement of the Gibbs draw, it must leave the block posterior invariant under $\pi_{\mathcal{B}}$. To see that our update satisfies this, we let $\rho^{\boldsymbol{y}}_{\tau}$ be the law of the interpolant eq. (3) when the endpoints $\boldsymbol{z}^{\partial\mathcal{B}}_{\mathbf{1}} = \boldsymbol{x}_{\partial\mathcal{B}}$ and the interior $\boldsymbol{z}^{\mathcal{B}}_{\mathbf{1}} \sim \pi_{\mathcal{B}}$. Under exact guidance, we know from section 2.3 that the guided SDE eq. (6) shares the same marginals as this interpolant. Thus, it transports $\rho^{\boldsymbol{y}}_{\tau}$ to $\rho^{\boldsymbol{y}}_{\mathbf{1}} = \pi_{\mathcal{B}}$, implying the invariance. For a formal proof of this, see proposition 1. We also note that similar observations have been made by Hill (2026); Kang et al. (2026).

Together with the overlapping blocks above, the DAWIS Block Smoother is a valid MCMC scheme with eq. (2) as its stationary distribution under perfect guidance (see corollary 1). At $\tau_{\min} = 0$, the noising discards the interior, recovering the Block Gibbs Smoother. Larger values of $\tau_{\min}$ act as a smaller step size, keeping the interior closer to its current value. This slows mixing, but shrinks the integration interval to $1 - \tau_{\min}$, making each cycle cheaper and exposing it to less guidance error. In the experiments, we initialize from the DAWIS Lagged Smoother and run a few sweeps with $\tau_{\min}$ close to one, refining an already assimilated trajectory rather than sampling it from scratch.

4 Related Works

Guiding forecast models. Many ML-based filters guide a learned forecast model towards the incoming observations, sampling from $p(\boldsymbol{x}_n \mid \boldsymbol{y}_n, \boldsymbol{x}_{n-1})$ (Chen et al., 2025; Savary et al., 2026) or from a longer window $p(\boldsymbol{x}_{n-w+1:n} \mid \boldsymbol{y}_{n-w+1:n}, \boldsymbol{x}_{n-w})$ (Shysheya et al., 2024). Following Shysheya et al. (2024) we refer to these as *Joint AR* $1|w$ and *Joint AR* $w|1$; both are instances of algorithm 1 with an empty inversion stage (table 1). A softer variant treats $\boldsymbol{x}_{n-w}$ as a trailing pseudo-observation rather than a hard anchor (Shysheya et al., 2024), used as the *SDA-Filter* baseline in Andrae et al. (2026).

Decoupled forecast-analysis steps. Classical methods such as the Kalman filter and cycled 3DVar target filtering by alternating forecast and analysis steps (Kalnay, 2002). DAISI (Andrae et al., 2026) adapts this to ML-based filters by coupling a forecast model with a stationary generative prior. DAWIS generalizes this inversion step to windows (section 3.1). Like the classical schemes, these assimilate only the latest observation, assuming past observations have already entered the estimate. 4DVar solves this by coupling several forecasting steps in a row, optimizing the initial state based on a window of observations (Kalnay, 2002). While extensions such as the Ensemble of Data Assimilations (EDA) with 4D-Var (Bonavita et al., 2012), referred to here as En4DVar, propagate an ensemble, the method still relies on a MAP estimate and requires inflation and perturbation to maintain ensemble spread.

All-at-once smoothing. Smoothing instead targets the full trajectory posterior $p(\boldsymbol{x}_{0:N}|\boldsymbol{y}_{1:N})$. Learning a flow-based prior over an entire trajectory is intractable, since the state dimension grows with N. Score-based Data Assimilation (*SDA*) (Rozet & Louppe, 2023) circumvents this by learning a prior only over local windows and coupling them at inference time, enabling parallel all-at-once sampling of trajectories of arbitrary length. The limited field of view comes at a cost: information propagates between distant states only through repeated local updates, so many solver steps and additional correction steps are needed before trajectories become coherent. With multimodal observations this can fail outright, as two ends of a trajectory may settle on incompatible modes. We compare SDA to DAWIS in the experiments.

Unified frameworks. Closest to our work are methods that assign an independent noise level to each state in a window, using a single model to cover several tasks. The idea originates in sequence generation, where rolling schedules noise later frames more heavily than earlier ones and the window slides forward during sampling (Ruhe et al., 2024; Cachay et al., 2025; Chen et al., 2024). Forcing-DAS (Jia et al., 2026) brings this construction to DA, learning a joint trajectory prior and selecting filtering, fixed-lag or full-sequence smoothing through the inference schedule. Its filter, however, is a conditional forecast that does not use the window to revise past states, and its full-sequence smoother requires a prior over the entire trajectory. Its fixed-lag smoother, *ForcingDAS-Pyr*, is closest to DAWIS, and we compare to a windowed version of it in our experiments. Like Joint AR $w|1$, it samples from $p(\boldsymbol{x}_{n-w+1:n} \mid \boldsymbol{y}_{n-w+1:n}, \boldsymbol{x}_{n-w})$, conditioning on a fixed $\boldsymbol{x}_{n-w}$. But, instead of running each slot all the way to a clean state, it keeps the window at rolling noise levels and denoises each slot one level per cycle. DAWIS shares the aim of one model for all tasks, but through inversion it revises all states in the window, including the nachor, and scales to full-trajectory smoothing without a costly full-sequence prior.

5 Experiments

We evaluate DAWIS on the SQG system and the `SEVIR` dataset with multiple observation operators for both filtering and smoothing. We present the CRPS for all experiments in table 2 and additional results in section E. We evaluate 10 trajectories with 20 ensemble members for each experiment and report both the mean and standard deviation. Our proposed method DAWIS consistently performs well across the whole range of experiments with different data and observation operators. Additional details about the experiments and the hyperparameters used is presented in section D.

Surface Quasi-Geostrophic (SQG) Dynamics. To evaluate DAWIS, we consider a Surface Quasi-Geostrophic (SQG) system, which provides a compact yet challenging setting for studying data assimilation in turbulent flows. The model evolves a scalar field θ through nonlinear advection together with forcing and dissipative processes, resulting in complex multiscale dynamics and strong sensitivity to the initial state (Tulloch & Smith, 2009). We use the SQG configuration from Andrae et al. (2026) to facilitate direct and fair comparison with previous results, discretizing the state on a 64×64 spatial grid and considering trajectories of 100 steps with a 3-hour interval between consecutive states. The observation scenarios, including the observation operators, noise levels, and spatial sparsity, are varied as specified in table 13. We use the numerical forward model for DAWIS Filter/Lagged Smoother and all other methods that require an external forward model. For further details of the underlying model and simulation setup we refer to Andrae et al. (2026).

Table 2: The CRPS for experiments on SQG and SEVIR. We display the mean and standard deviation across 10 independent trajectories, averaged over the last 20 (10 for SEVIR) steps. The best score for each experiment is highlighted in **bold** and the second best with an underline. Smoothing methods calculate the metrics over the whole trajectory.

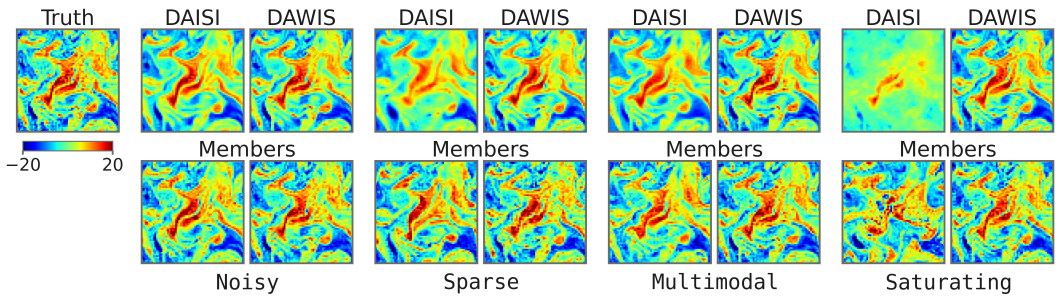
Method	Noisy	Sparse	Multimodal	Saturating	SEVIR
Filtering					
EnSF Bao et al. (2024)	4.95±0.83	4.11±0.61	1.79±1.38	5.79±1.72	21.92±20.35
FlowDAS Chen et al. (2025)	2.73±0.50	3.22±0.56	2.50±0.48	3.66±0.41	15.66±7.17
LETKF Liang et al. (2025)	1.32±0.14	2.22±0.25	4.86±1.11	2.28±1.06	4.67±3.17
DAISI Andrae et al. (2026)	1.30±0.14	1.69±0.20	1.55±0.09	1.79±0.45	3.97±2.65
SDA-Filter Andrae et al. (2026)*	1.32±0.13	1.75±0.25	2.03±0.32	4.26±0.70	3.79±2.25
Joint AR 1|W Shysheya et al. (2024)	3.83±0.41	4.01±0.54	3.97±0.41	3.77±0.44	6.42±3.86
Joint AR W|1 Shysheya et al. (2024)	1.27±0.14	1.62±0.15	1.21±0.10	1.20±0.16	4.28±2.73
DAWIS Filter	**1.03**±0.10	**1.22**±0.14	**1.04**±0.08	**0.86**±0.13	3.86±2.12
DAWIS-Joint Filter	1.18±0.13	1.26±0.16	1.21±0.13	0.99±0.16	**3.58**±2.30
Fixed-lag Smoothing					
SDA-Filter	1.37±0.08	1.54±0.15	1.75±0.24	3.32±0.56	5.08±2.83
ForcingDAS-Pyr Jia et al. (2026)	1.36±0.08	1.28±0.11	1.05±0.09	1.09±0.13	3.60±2.53
Joint AR W|1 Shysheya et al. (2024)	1.13±0.06	1.29±0.06	1.06±0.05	1.05±0.10	3.59±2.61
DAWIS Lagged Smoother	**0.93**±0.05	**1.05**±0.06	**0.94**±0.04	**0.76**±0.06	3.36±2.03
DAWIS-Joint Lagged Smoother	1.07±0.06	1.08±0.06	1.09±0.06	0.83±0.07	**3.34**±2.07
Joint Smoothing					
SDA Rozet & Louppe (2023)	1.09±0.07	1.05±0.06	2.74±0.35	1.15±0.07	3.78±2.57
Block Gibbs Smoother	0.92±0.06	0.99±0.07	0.90±0.05	0.86±0.09	3.58±2.35
DAWIS Block Smoother	**0.90**±0.05	**0.98**±0.06	**0.89**±0.05	**0.68**±0.06	**3.18**±2.17

Precipitation Nowcasting using SEVIR. We also evaluate DAWIS on the SEVIR dataset (Veillette et al., 2020), which consists of vertically integrated liquid radar observations of convective storms over the United States. Following the experimental setup of Chen et al. (2025), we use 384×384 spatial fields at $3\,\mathrm{km}$ resolution with observations every $10\,\mathrm{min}$. We adopt their linear Gaussian observation setting with $10\,\%$ spatial coverage and use the corresponding pretrained forecasting model for experiments that require an external forecast. Further experimental details are given by Chen et al. (2025). We use the FlowDAS (Chen et al., 2025) forward model for all methods that require an external forward model.

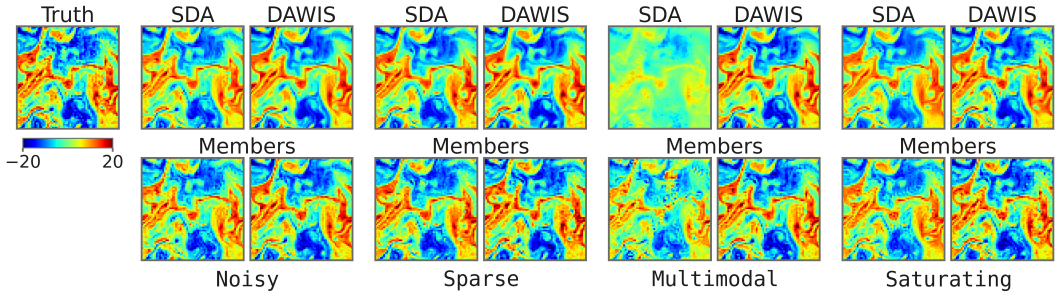
Filtering. DAWIS Filter has the lowest CRPS and RMSE of all filters in the four SQG settings, and DAWIS-Joint Filter is second in most of them (tables 2 and 14 and figs. 4 to 6). On SEVIR, DAWIS-Joint Filter is the best filter and also improves on DAWIS Filter, which relies on the pretrained FlowDAS forecast model, showing that the forecast generated by the window prior is competitive with that of a dedicated forecast model. While the DAWIS models are robust across all experiments, several baselines break down in individual settings. LETKF struggles on Multimodal, where a Gaussian update cannot represent the bimodal posterior (fig. 25); SDA-Filter fails on Saturating (fig. 26); EnSF and FlowDAS do not reproduce the high-frequency content of the fields (fig. 20).

Fixed-lag Smoothing. Both DAWIS/DAWIS-Joint Lagged Smoothers improve on the corresponding filters in every setting (tables 2 and 14 and figs. 8 and 9). Since the two estimates come from the same assimilation cycle, the smoother requires no additional computation. DAWIS Lagged Smoother is the best fixed-lag smoother on all four SQG settings, and on SEVIR DAWIS-Joint Lagged Smoother has the lowest CRPS with DAWIS Lagged Smoother close behind.

Joint Smoothing. The block smoothers are initialized from the DAWIS Lagged Smoother for SQG and ther DAWIS-Joint Lagged Smoother for SEVIR and refine the trajectory with alternating forward and backward sweeps. The (approximative guidance-based) Block Gibbs Smoother, which regenerates each block from pure noise, gives small gains on Noisy, Sparse, and Multimodal but makes the initialization worse on Saturating and SEVIR (tables 2 and 14 and figs. 4 and 5).



(a) **Filtering:** DAISI vs DAWIS Filter



(b) **Smoothing:** SDA vs DAWIS Block Smoother

Figure 2: Ensemble mean and a single member for all SQG experiments. Here visualized for the endpoint for filtering and the midpoint for smoothing. DAWIS exhibits robust performance for all settings, including the challenging multimodal and saturating observations where DAISI and SDA struggle in comparison.

We attribute this to the guidance error accumulating over the full regeneration path, whereas a turning point close to one keeps each update small. The DAWIS Block Smoother improves on the initialization in all settings, including `Saturating` and `SEVIR`, and has the lowest CRPS and RMSE of all methods on all experiments.

6 Conclusion

We presented DAWIS, which treats data assimilation as inversion and guided regeneration of a window of states under a multitask trajectory prior. With a separate flow time for each state, the same trained model can perform filtering, fixed-lag smoothing, or block smoothing, with the task determined solely by the turning-point vector rather than retraining. The forecast can also be drawn from the prior itself, which removes the dependence on an external forecast model at a small cost in accuracy. Across multiple observation operators on SQG and on `SEVIR`, DAWIS demonstrates competitive skill and calibration for filtering, fixed lag smoothing, and block smoothing.

Limitations & Future Work. Our evaluation covers a 64×64 SQG system and the `SEVIR` dataset with synthetic observations, a simplified setting compared to the state dimension of operational systems and without the heterogeneous, imperfectly known operators of a real observing network. Scaling DAWIS to higher-dimensional states and real observations is the natural next step. Since the framework assumes nothing about the underlying dynamics, we also expect it to transfer to other domains where windowed generative priors are natural, such as computational fluid dynamics and video. A second direction is sampling cost: each cycle inverts and regenerates a window of W states using 100 solver steps, which could potentially be reduced using distillation approaches such as consistency models or flow maps. Finally, DAWIS inherits the limitations of whichever approximation of $p(\boldsymbol{y}|\boldsymbol{z}_{\tau_t})$ it is paired with. In our experiments, we use MMPS (Rozet et al., 2024), but more accurate guidance methods should transfer directly to DAWIS and improve the performance.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We have additionally used AI to formulate mathematical claims, provide critical ingredients for proving mathematical claims and assist in the writing of proofs. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI tools to edit the research paper to improve readability and to assist software development. We have reviewed all AI-assisted work. LLM-generated code and text suggestions were verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We have no ethical concerns or conflicts of interest to declare.

REPRODUCIBILITY STATEMENT

We have taken several steps to make our results reproducible. The DAWIS method is fully specified in algorithm 1, with the design choices that recover each filter and smoother given in table 1 and described in sections 3.1 to 3.3. The notation is collected in section A, and the multitask interpolant and the MMPS guidance, including the relation between drift, score and the learned conditional expectations, are derived in sections B.1 and B.2. The theoretical results in section C state their assumptions (A1)–(A3) explicitly and come with complete proofs. Section D gives the details needed to rerun our experiments: the DAWIS hyperparameters and how they were tuned (tables 3 and 4), the backbone architecture and optimizer settings (table 5), the configuration and hyperparameters of every baseline (tables 6 to 12), and the data generation, train/validation/test splits, normalization and observation operators for SQG and `SEVIR` (table 13 and fig. 3). All results are reported as the mean and standard deviation over 10 independent trajectories with 20 ensemble members (tables 2, 14 and 15). The source code for training, assimilation and evaluation, together with the configuration files for all experiments, will be released upon acceptance.

REFERENCES

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Martin Andrae, Erik Wikingsson, So Takao, Tomas Landelius, and Fredrik Lindsten. DAISI: Data assimilation with inverse sampling using stochastic interpolants. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=OWz1n5HgcC`.

Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data Assimilation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016. doi: 10.1137/1.9781611974546.

Ross N Bannister. A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703):607–633, 2017.

Feng Bao, Zezhong Zhang, and Guannan Zhang. An ensemble score filter for tracking high-dimensional nonlinear dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 432:117447, 2024.

Thomas Bengtsson, Peter Bickel, and Bo Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, volume 2, pp. 316–335. Institute of Mathematical Statistics, 2008.

Massimo Bonavita, Lars Isaksen, and Elías Hólm. On the use of eda background error variances in the ecmwf 4d-var. *Quarterly Journal of the Royal Meteorological Society*, 138(667):1540–1559, 2012. doi: https://doi.org/10.1002/qj.1899. URL `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.1899`.

Salva Rühling Cachay, Miika Aittala, Karsten Kreis, Noah D Brenowitz, Arash Vahdat, Morteza Mardani, and Rose Yu. Elucidated rolling diffusion models for probabilistic forecasting of complex dynamics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Edoardo Calvello, Pierre Monmarché, Andrew M Stuart, and Urbain Vaes. Accuracy of the ensemble Kalman filter in the near-linear setting. *arXiv preprint arXiv:2409.09800*, 2024.

Chris K Carter and Robert Kohn. On gibbs sampling for state space models. *Biometrika*, 81(3): 541–553, 1994.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Siyi Chen, Yixuan Jia, Qing Qu, He Sun, and Jeffrey A Fessler. FlowDAS: A stochastic interpolant-based framework for data assimilation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Florentin Coeurdoux, Nicolas Dobigeon, and Pierre Chainais. Plug-and-play split gibbs sampler: embedding deep generative priors in bayesian inference. *IEEE Transactions on Image Processing*, 33:3496–3507, 2024.

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010.

Mitch Hill. Exact global mcmc with denoising diffusion. *arXiv preprint arXiv:2609.00279*, 2026.

Yixuan Jia, Siyi Chen, Yida Pan, Xiao Li, Lianghe Shi, Chanyong Jung, Haijie Yuan, Ismail Alkhouri, Yue Cynthia Wu, Saiprasad Ravishankar, Jeffrey A Fessler, and Qing Qu. Forcing-das: Unified and robust data assimilation via diffusion forcing, 2026. URL `https://arxiv.org/abs/2605.14285`.

Eugenia Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 2002.

Hyunmo Kang, Noam Itzhak Levi, Corinna Elena Wegner, Daniel J Korchinski, and Matthieu Wyart. Sampling data with chains of forward-backward diffusion steps. *arXiv preprint arXiv:2605.27006*, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

Siming Liang, Hoang Tran, Feng Bao, Hristo G Chipilski, Peter Jan van Leeuwen, and Guannan Zhang. Ensemble score filter with image inpainting for data assimilation in tracking surface quasi-geostrophic dynamics with partial observations. *arXiv preprint arXiv:2501.12419*, 2025.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in Adam. *CoRR*, abs/1711.05101, 2017a.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017b.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

Christian A Naesseth, Fredrik Lindsten, Thomas B Schön, et al. Elements of Sequential Monte Carlo. *Foundations and Trends in Machine Learning*, 12(3):307–392, 2019.

Hugo Negrel, Florentin Coeurdoux, Michael Samuel Albergo, and Eric Vanden-Eijnden. Multitask learning with stochastic interpolants. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jimmy Olsson, Olivier Cappé, Randal Douc, and Éric Moulines. Sequential monte carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008. ISSN 13507265.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael Samuel Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=K5gV8Yptne.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

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

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 87647–87682. Curran Associates, Inc., 2024.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42818–42835. PMLR, 21–27 Jul 2024.

Thomas Savary, François Rozet, and Gilles Louppe. Training-free bayesian filtering with generative emulators. *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Neil Shephard and Michael K Pitt. Likelihood analysis of non-gaussian measurement time series. *Biometrika*, 84(3):653–667, 1997.

Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard E. Turner, and Emile Mathieu. On conditional diffusion models for PDE simulations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=nQl8EjyMzh.

Sumeetpal S Singh, Fredrik Lindsten, and Eric Moulines. Blocking strategies and stability of particle gibbs samplers. *Biometrika*, 104(4):953–969, 2017.

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

Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pp. 1701–1728, 1994.

Ross Tulloch and K Shafer Smith. Quasigeostrophic turbulence with explicit surface dynamics: Application to the atmospheric energy spectrum. *Journal of the atmospheric sciences*, 66(2): 450–467, 2009.

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

Zihui Wu, Yu Sun, Yifan Chen, Bingliang Zhang, Yisong Yue, and Katherine L Bouman. Principled probabilistic imaging using diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 37:118389–118427, 2024.

A NOTATION

For reference, we collect the notation used throughout the paper. We keep physical (assimilation) time, indexed by n, separate from the flow time τ of the generative model. Subscripts index physical time and superscripts index positions within a window.

Indices and physical time.

- n: physical (assimilation) time index; N: final time index.
- w: window length, so a window holds $w+1$ states; $k = 0, \dots, w$: slot (position) within the window.
- $\boldsymbol{x}_n \in \mathbb{R}^d$: physical state at time n; d its dimension.
- $\boldsymbol{x}_n^{\mathrm{f}}$: forecast of $\boldsymbol{x}_n$ from an external forecasting model.
- $\hat{\boldsymbol{x}}_n \in \mathbb{R}^d$: current sample (analysis) of $\boldsymbol{x}_n$ produced by an earlier cycle
- $\boldsymbol{x}_{n:n+w} := (\boldsymbol{x}_n, \dots, \boldsymbol{x}_{n+w})$: state trajectory over a window; $D = (w+1)d$ its dimension.
- $\mathcal{W} = \{n, \dots, n+w\}$: the set of time indices in a window; $\boldsymbol{x}_{\mathcal{W}}$, $\boldsymbol{y}_{\mathcal{W}}$, $\hat{\boldsymbol{x}}_{\mathcal{W}}$: the states, observations and updated estimate over it.
- $\boldsymbol{y}_n$: observation at time n, with likelihood $p(\boldsymbol{y}_n \mid \boldsymbol{x}_n)$.
- $\mathcal{B}$: the interior of a block; $\partial\mathcal{B} \subseteq \{n, n+w\}$: its boundary, the window endpoints held fixed; $\boldsymbol{y}_{\mathcal{B}}$: observations of the interior states.
- J: ensemble size; $j = 1, \dots, J$: ensemble member index.

Flow time and the multitask schedule.

- $\boldsymbol{\tau} \in [0,1]^D$: flow-time vector, one entry per component; $\tau = 0$ is noise, $\tau = 1$ is data. In DAWIS, $\boldsymbol{\tau}$ is constant within each state and we write $\boldsymbol{\tau} = (\tau^0, \dots, \tau^w)$ for the $w+1$ free flow times.
- $t \in [0,1]$: path parameter, always running from 0 to 1; not itself a flow time.
- $\boldsymbol{\tau}_t \in C^2([0,1]; [0,1]^D)$: the path along which the flow times evolve during sampling; $\dot{\boldsymbol{\tau}}_t$ its time derivative.
- $\boldsymbol{\tau}_{\min} = (\tau_{\min}^0, \dots, \tau_{\min}^w)$: turning points, the flow times to which each slot is inverted before regeneration; $\tau_{\min}$: the common turning point of the interior slots in the block smoother.
- $\alpha_{\boldsymbol{\tau}}, \beta_{\boldsymbol{\tau}}$: interpolant coefficients, evaluated elementwise; α', β': their derivatives with respect to flow time.
- $\boldsymbol{z_0} \sim \rho_{\boldsymbol{0}}$, $\boldsymbol{z_1} \sim \rho_{\boldsymbol{1}}$: latent (noise) and data endpoints of the interpolant.
- $\boldsymbol{z}_{\boldsymbol{\tau}}$: multitask interpolant eq. (3); $\boldsymbol{z}_{\boldsymbol{\tau}_t}$ its value along a path. Superscripts select slots, e.g. $\boldsymbol{z}_{\boldsymbol{\tau}}^k$, $\boldsymbol{z}_{\boldsymbol{\tau}}^{\mathcal{B}}$, $\boldsymbol{z}_{\boldsymbol{1}}^{\partial\mathcal{B}}$.
- $\rho_{\boldsymbol{\tau}}$, $p_{\boldsymbol{\tau}}$: law and density of $\boldsymbol{z}_{\boldsymbol{\tau}}$; $\rho_{\boldsymbol{\tau}}^{\boldsymbol{y}}$: the same for the posterior interpolant, with $\rho_{\boldsymbol{1}}^{\boldsymbol{y}}$ the posterior target under guidance.
- $\boldsymbol{\varepsilon}_{\boldsymbol{\tau}_t} \geq \boldsymbol{0}$: diffusion coefficient of the generative SDE; W_t: Brownian motion.
- $\odot$: elementwise multiplication.

Distributions.

- $\pi_n := p(\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n})$: filtering distribution.
- $\pi_n^\ell := p(\boldsymbol{x}_n \mid \boldsymbol{y}_{1:n+\ell})$: fixed-lag smoothing distribution with ℓ steps of look-ahead.
- $p(\boldsymbol{x}_{0:N} \mid \boldsymbol{y}_{1:N})$: joint smoothing distribution.
- $\pi_{\mathcal{B}} := p(\boldsymbol{x}_{\mathcal{B}} \mid \boldsymbol{x}_{\partial\mathcal{B}}, \boldsymbol{y}_{\mathcal{B}})$: block posterior.

B Details on Multitask Stochastic Interpolants

This appendix covers the multitask stochastic interpolant framework that DAWIS is built on. Section B.1 shows that the drift and score along any flow-time path follow from the same two learned conditional expectations. This is why a single network can be used for filtering, fixed-lag smoothing and block smoothing without retraining. Section B.2 extends these quantities to posterior sampling and gives the MMPS approximation of the likelihood score, which we use in all DAWIS experiments.

B.1 Multitask Interpolants

Recall the multitask interpolant

$$\boldsymbol{z}_{\boldsymbol{\tau}} = \alpha_{\boldsymbol{\tau}} \odot \boldsymbol{z}_{\mathbf{0}} + \beta_{\boldsymbol{\tau}} \odot \boldsymbol{z}_{\mathbf{1}}, \qquad \boldsymbol{z}_{\mathbf{0}} \sim \rho_{\mathbf{0}},\ \boldsymbol{z}_{\mathbf{1}} \sim \rho_{\mathbf{1}}, \qquad \boldsymbol{\tau} \in [0,1]^D, \tag{10}$$

and the conditional expectations

$$\boldsymbol{\eta}_{\mathbf{0}}(\boldsymbol{\tau}, \boldsymbol{z}) := \mathbb{E}[\boldsymbol{z}_{\mathbf{0}} \mid \boldsymbol{z}_{\boldsymbol{\tau}} = \boldsymbol{z}], \qquad \boldsymbol{\eta}_{\mathbf{1}}(\boldsymbol{\tau}, \boldsymbol{z}) := \mathbb{E}[\boldsymbol{z}_{\mathbf{1}} \mid \boldsymbol{z}_{\boldsymbol{\tau}} = \boldsymbol{z}]. \tag{11}$$

Both are functions of the point $\boldsymbol{\tau}$ alone and do not depend on the path taken to reach it. Let $\boldsymbol{\tau}_t \in C^2([0,1])^D$ be a path and let $\rho_{\mathbf{0}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then the drift and score of the interpolant eq. (10) along $\boldsymbol{\tau}_t$ are given by

$$\boldsymbol{b}_{\boldsymbol{\tau}}(t, \boldsymbol{z}) = \dot{\boldsymbol{\tau}}_t \odot \Big(\alpha'_{\boldsymbol{\tau}_t} \odot \boldsymbol{\eta}_{\mathbf{0}}(\boldsymbol{\tau}_t, \boldsymbol{z}) + \beta'_{\boldsymbol{\tau}_t} \odot \boldsymbol{\eta}_{\mathbf{1}}(\boldsymbol{\tau}_t, \boldsymbol{z}) \Big), \tag{12}$$

$$\boldsymbol{s}_{\boldsymbol{\tau}}(t, \boldsymbol{z}) = -\alpha_{\boldsymbol{\tau}_t}^{-1} \odot \boldsymbol{\eta}_{\mathbf{0}}(\boldsymbol{\tau}_t, \boldsymbol{z}), \tag{13}$$

where α', β' denote derivatives of the scalar curves with respect to their own argument and all operations are elementwise.

Two consequences are worth emphasizing. First, eq. (12) and eq. (13) depend on the path only through the factor $\dot{\boldsymbol{\tau}}_t$ and through the point $\boldsymbol{\tau}_t$ at which $\boldsymbol{\eta}_{\mathbf{0}}, \boldsymbol{\eta}_{\mathbf{1}}$ are evaluated. A single model therefore serves every path, and no retraining is required when the schedule changes. Second, the initial condition is the law of the interpolant at $\boldsymbol{\tau}_0$ rather than pure noise, so a partially noised sample may be used as the starting point; this is what makes the inversion step of algorithm 1 possible.

In practice we learn $\boldsymbol{\eta}_{\mathbf{0}}, \boldsymbol{\eta}_{\mathbf{1}}$ by minimizing

$$\mathcal{L}_i(\theta) = \mathbb{E}_{\boldsymbol{z}_{\mathbf{0}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),\, \boldsymbol{z}_{\mathbf{1}} \sim \rho_{\mathbf{1}},\, \boldsymbol{\tau} \sim \mathcal{U}} \Big[\big\| \boldsymbol{\eta}_i^{\theta}(\boldsymbol{\tau}, \boldsymbol{z}_{\boldsymbol{\tau}}) - \boldsymbol{z}_i \big\|^2 \Big], \qquad i \in \{\mathbf{0}, \mathbf{1}\}, \tag{14}$$

whose minimizers are exactly the conditional expectations eq. (11). In DAWIS the components of $\boldsymbol{\tau}$ are constant within each state, so the training distribution $\mathcal{U}$ draws $w+1$ independent times $\tau^k \sim \mathcal{U}[0,1]$, one per position in the window, and assigns τ^k to every component of state k. We predict $\boldsymbol{\eta}_{\mathbf{0}}$ and $\boldsymbol{\eta}_{\mathbf{1}}$ as separate output channels of a single network; see Appendix D.2 for details.

B.2 Guidance

Replacing the data distribution $\rho_{\mathbf{1}}$ by the posterior $\rho_{\mathbf{1}}^{\boldsymbol{y}}$ leaves the conditional structure of the interpolant unchanged, so the conditional score and drift differ from their unconditional counterparts only through the likelihood score (see Andrae et al. (2026) for a derivation):

$$\boldsymbol{s}^{\boldsymbol{y}}(\boldsymbol{\tau}_t, \boldsymbol{z}) = \boldsymbol{s}(\boldsymbol{\tau}_t, \boldsymbol{z}_{\boldsymbol{\tau}_t}) + \nabla_{\boldsymbol{z}_{\boldsymbol{\tau}_t}} \log p(\boldsymbol{y} \mid \boldsymbol{z}_{\boldsymbol{\tau}_t}), \tag{15}$$

$$\boldsymbol{b}^{\boldsymbol{y}}(\boldsymbol{\tau}_t, \boldsymbol{z}_{\boldsymbol{\tau}_t}) = \boldsymbol{b}(\boldsymbol{\tau}_t, \boldsymbol{z}_{\boldsymbol{\tau}_t}) + \boldsymbol{\lambda}_{\boldsymbol{\tau}_t} \odot \nabla_{\boldsymbol{z}_{\boldsymbol{\tau}_t}} \log p(\boldsymbol{y} \mid \boldsymbol{z}_{\boldsymbol{\tau}_t}), \tag{16}$$

where the drift is rescaled componentwise by

$$\boldsymbol{\lambda}_{\boldsymbol{\tau}_t} := \frac{\alpha_{\boldsymbol{\tau}_t} \odot \gamma_{\boldsymbol{\tau}_t}}{\beta_{\boldsymbol{\tau}_t}}, \qquad \gamma_{\boldsymbol{\tau}} := \dot{\boldsymbol{\tau}}_t \odot \left(\beta'_{\boldsymbol{\tau}} \odot \alpha_{\boldsymbol{\tau}} - \alpha'_{\boldsymbol{\tau}} \odot \beta_{\boldsymbol{\tau}} \right). \tag{17}$$

Substituting eq. (15) into eq. (4) gives the guided SDE eq. (6) of the main text, whose flow transports $\rho_{\boldsymbol{\tau}_0}$ to the posterior $\rho_{\mathbf{1}}^{\boldsymbol{y}}$.

For the linear interpolant used throughout, $\alpha(\tau) = 1 - \tau$ and $\beta(\tau) = \tau$, we have $\beta'_\tau \alpha_\tau - \alpha'_\tau \beta_\tau \equiv 1$ and eq. (17) reduces to $\lambda_t^i = \dot{\tau}_t^i (1 - \tau_t^i)/\tau_t^i$. Two properties are worth noting. A component held

fixed has $\dot{\tau}_t^i = 0$ and therefore receives no guidance, as it should, since it is treated as known. Conversely, λ_t^i grows without bound as $\tau_t^i \to 0$, so in practice we clamp $\beta_{\boldsymbol{\tau}_t}$ away from zero.

Exact conditional sampling still requires the likelihood score $\nabla_{\boldsymbol{z}} \log p(\boldsymbol{y} \mid \boldsymbol{z}_{\boldsymbol{\tau}_t} = \boldsymbol{z})$, which is for the most part analytically intractable and requires approximation. Below we present MMPS, one such approximation. In addition we multiply the likelihood score by a guidance strength $\zeta > 0$, tuned per problem setup.

Throughout, let $\mathbf{A}_t := \mathrm{diag}(\alpha_{\boldsymbol{\tau}_t})$ and $\mathbf{B}_t := \mathrm{diag}(\beta_{\boldsymbol{\tau}_t})$ denote the interpolant coefficients of the current schedule point, and recall from Appendix B.1 that the learned network provides $\boldsymbol{\eta}_1(\boldsymbol{\tau}_t, \boldsymbol{z}) \approx \mathbb{E}[\boldsymbol{z}_1 \mid \boldsymbol{z}_{\boldsymbol{\tau}_t} = \boldsymbol{z}]$. Since the prior is defined over a window, $\boldsymbol{y}$ collects the observations $\boldsymbol{y}_{n-w:n}$ of that window and $\mathcal{H}$ acts blockwise, one observation operator per state.

B.2.1 Moment-Matching Posterior Sampling (MMPS)

MMPS approximates the denoising posterior by a Gaussian,

$$p(\boldsymbol{z}_1 \mid \boldsymbol{z}_{\boldsymbol{\tau}_t} = \boldsymbol{z}) \approx \mathcal{N}\big(\boldsymbol{\mu}_t(\boldsymbol{z}),\, \boldsymbol{\Sigma}_t(\boldsymbol{z})\big), \qquad \boldsymbol{\mu}_t(\boldsymbol{z}) := \boldsymbol{\eta}_1(\boldsymbol{\tau}_t, \boldsymbol{z}), \tag{18}$$

whose covariance $\boldsymbol{\Sigma}_t(\boldsymbol{z}) := \mathbb{E}[\boldsymbol{z}_1 \boldsymbol{z}_1^\top \mid \boldsymbol{z}] - \boldsymbol{\mu}_t \boldsymbol{\mu}_t^\top$ is available from the Jacobian of the mean (Rozet et al., 2024):

$$\boldsymbol{\Sigma}_t(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \boldsymbol{\mu}_t(\boldsymbol{z})\, \mathbf{A}_t^2 \mathbf{B}_t^{-1}. \tag{19}$$

For a single shared flow time, $\mathbf{A}_t^2 \mathbf{B}_t^{-1}$ reduces to the scalar α_t^2/β_t and eq. (19) recovers the usual expression; with per-state flow times the factor is a diagonal matrix that does not commute with $\nabla_{\boldsymbol{z}} \boldsymbol{\mu}_t$, so the right-multiplication matters.

Given a Gaussian likelihood $p(\boldsymbol{y} \mid \boldsymbol{z}_1) = \mathcal{N}(\boldsymbol{y} \mid \mathcal{H}(\boldsymbol{z}_1), \sigma_{\boldsymbol{y}}^2 \mathbf{I})$, marginalizing the Gaussian approximation yields

$$\nabla_{\boldsymbol{z}} \log p(\boldsymbol{y} \mid \boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log \left(\int_{\mathbb{R}^D} p(\boldsymbol{y} \mid \boldsymbol{z}_1)\, p(\boldsymbol{z}_1 \mid \boldsymbol{z}) \mathrm{d}\boldsymbol{z}_1 \right) \tag{20}$$

$$\approx \nabla_{\boldsymbol{z}} \log \left(\int_{\mathbb{R}^D} p(\boldsymbol{y} \mid \boldsymbol{z}_1)\, \mathcal{N}\big(\boldsymbol{z}_1 \mid \boldsymbol{\mu}_t(\boldsymbol{z}), \boldsymbol{\Sigma}_t(\boldsymbol{z})\big) \mathrm{d}\boldsymbol{z}_1 \right) \tag{21}$$

$$\approx \big(\nabla_{\boldsymbol{z}} \boldsymbol{\mu}_t\big)^\top \mathbf{H}_t^\top \left(\sigma_{\boldsymbol{y}}^2 \mathbf{I} + \mathbf{H}_t\, \nabla_{\boldsymbol{z}} \boldsymbol{\mu}_t\, \mathbf{A}_t^2 \mathbf{B}_t^{-1} \mathbf{H}_t^\top \right)^{-1} \big(\boldsymbol{y} - \mathcal{H}(\boldsymbol{\mu}_t)\big), \tag{22}$$

where $\mathbf{H}_t := \nabla_{\boldsymbol{z}_1} \mathcal{H}(\boldsymbol{z}_1)\big|_{\boldsymbol{z}_1 = \boldsymbol{\mu}_t(\boldsymbol{z})}$. The final step neglects the dependence of $\boldsymbol{\Sigma}_t$ on $\boldsymbol{z}$; the marginalization itself is exact whenever $\mathcal{H}$ is linear, though eq. (22) remains an approximation of the true likelihood score because $p(\boldsymbol{z}_1 \mid \boldsymbol{z}_{\boldsymbol{\tau}_t})$ is not Gaussian in general.

In practice neither $\nabla_{\boldsymbol{z}} \boldsymbol{\mu}_t$ nor the inverse is formed explicitly: the solve is performed with conjugate gradients, requiring only Jacobian-vector products with $\boldsymbol{\eta}_1$.

C Theorems

Throughout, we assume (A1) $\rho_{\mathbf{0}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and interpolant coefficients α, β as in section 2.2 with $\alpha, \beta > 0$ on $(0, 1)$. , with all operations elementwise; (A2) the conditional expectations $\boldsymbol{\eta}_{\mathbf{0}}, \boldsymbol{\eta}_{\mathbf{1}}$ in eq. (5) and the likelihood score $\nabla_{\boldsymbol{z}} \log p(\boldsymbol{y} \mid \boldsymbol{z}_{\boldsymbol{\tau}} = \boldsymbol{z})$ are exact; and (A3) the diffusion coefficient $\boldsymbol{\varepsilon}_{\boldsymbol{\tau}_t} \geq \mathbf{0}$ vanishes on every slot held at flow time 1, so that frozen slots stay frozen (the score $-\alpha_{\boldsymbol{\tau}}^{-1} \odot \boldsymbol{\eta}_{\mathbf{0}}$ is singular at $\alpha = 0$). The linear interpolant $\alpha_{\boldsymbol{\tau}} = \mathbf{1} - \boldsymbol{\tau}$, $\beta_{\boldsymbol{\tau}} = \boldsymbol{\tau}$ used in the paper satisfies (A1) with $\beta' \alpha - \alpha' \beta \equiv 1$.

For the block smoother we allow the boundary $\partial\mathcal{B} \subseteq \{n, n + w\}$ of a window $\{n, \dots, n + w\}$ to contain only the endpoints that have a neighbour outside the window, so that the two edge blocks $n = 0$ and $n + w = N$ have a single boundary slot and $\boldsymbol{x}_0$, $\boldsymbol{x}_N$ can be updated; eq. (8) holds verbatim for such blocks by the Markov property of eq. (1).

Proposition 1 (Invariance of the block update)**.** *Fix a block with interior $\mathcal{B}$ and boundary $\partial\mathcal{B}$, and let $\pi_{\mathcal{B}} = p(\boldsymbol{x}_{\mathcal{B}} \mid \boldsymbol{x}_{\partial\mathcal{B}}, \boldsymbol{y}_{\mathcal{B}})$ as in eq.* (8)*. Let $K_{\tau_{\min}}$ be the DAWIS Block Smoother update, i.e. algorithm 1 with $\boldsymbol{\tau}_0 = \boldsymbol{\tau}_{\min}$ equal to 1 on $\partial\mathcal{B}$ and $\tau_{\min}$ on $\mathcal{B}$: the initialization eq.* (9) *with $\boldsymbol{z}_{\mathbf{0}} \sim \rho_{\mathbf{0}}$*

independent of $\boldsymbol{x}_{\mathcal{B}}$, followed by the guidance stage, the guided SDE eq. (6) *with likelihood $p(\boldsymbol{y}_{\mathcal{B}} \mid \cdot)$ run along any path from $\boldsymbol{\tau}_{\min}$ to $\mathbf{1}$ that keeps $\partial\mathcal{B}$ at flow time* 1*. Then under* (A1)–(A3)*, for every $\tau_{\min} \in [0, 1]$,*

$$\pi_{\mathcal{B}} K_{\tau_{\min}} = \pi_{\mathcal{B}}, \tag{23}$$

where πK denotes the law of $\boldsymbol{x}' \sim K(\boldsymbol{x}, \cdot)$ for $\boldsymbol{x} \sim \pi$. At $\tau_{\min} = 0$, $K_0(\boldsymbol{x}_{\mathcal{B}}, \cdot) = \pi_{\mathcal{B}}$ for every $\boldsymbol{x}_{\mathcal{B}}$, which is the Block Gibbs update. At $\tau_{\min} = 1$, K_1 is the identity.

Proof. Consider the interpolant eq. (10) on the window with the boundary slots held at flow time 1, $\boldsymbol{z}_{\mathbf{1}}^{\partial\mathcal{B}} = \boldsymbol{x}_{\partial\mathcal{B}}$, and interior endpoint $\boldsymbol{z}_{\mathbf{1}}^{\mathcal{B}} \sim \pi_{\mathcal{B}}$. Write $\rho_\tau^{\boldsymbol{y}}$ for the law of its interior at common flow time τ, so that $\rho_1^{\boldsymbol{y}} = \pi_{\mathcal{B}}$ and $\rho_0^{\boldsymbol{y}} = \rho_{\mathbf{0}}$.

(a) Since $\alpha_1 = 0$ and $\beta_1 = 1$, the boundary slots satisfy $\boldsymbol{z}_{\boldsymbol{\tau}}^{\partial\mathcal{B}} = \boldsymbol{z}_{\mathbf{1}}^{\partial\mathcal{B}}$, so conditioning on $\boldsymbol{z}_{\boldsymbol{\tau}}$ with $\tau^{\partial\mathcal{B}} = 1$ conditions on $\boldsymbol{x}_{\partial\mathcal{B}}$: on the interior, $\boldsymbol{\eta}_{\mathbf{0}}, \boldsymbol{\eta}_{\mathbf{1}}$ are the conditional expectations of the interpolant with data law $p(\boldsymbol{x}_{\mathcal{B}} \mid \boldsymbol{x}_{\partial\mathcal{B}})$. As $\pi_{\mathcal{B}} \propto p(\boldsymbol{x}_{\mathcal{B}} \mid \boldsymbol{x}_{\partial\mathcal{B}})\, p(\boldsymbol{y}_{\mathcal{B}} \mid \boldsymbol{x}_{\mathcal{B}})$, the interpolant above is the posterior interpolant of this conditional prior, and by section B.2 its drift and score are the guided quantities eq. (15), which are exact under (A2). By Negrel et al. (2025) the SDE preserves its marginals for every admissible $\boldsymbol{\varepsilon}$, and by (A3) the boundary does not move. Started from $\rho_{\boldsymbol{\tau}_{\min}}^{\boldsymbol{y}}$, the guidance stage thus returns a sample from $\rho_1^{\boldsymbol{y}} = \pi_{\mathcal{B}}$.

(b) If $\boldsymbol{x}_{\mathcal{B}} \sim \pi_{\mathcal{B}}$, the initialization returns $\alpha_{\tau_{\min}} \boldsymbol{z}_{\mathbf{0}} + \beta_{\tau_{\min}} \boldsymbol{x}_{\mathcal{B}}$ with $\boldsymbol{z}_{\mathbf{0}} \sim \rho_{\mathbf{0}}$ independent of $\boldsymbol{x}_{\mathcal{B}}$, which is by definition a draw from $\rho_{\boldsymbol{\tau}_{\min}}^{\boldsymbol{y}}$.

Composing (b) and (a), $K_{\tau_{\min}}$ maps a sample from $\pi_{\mathcal{B}}$ to a sample from $\pi_{\mathcal{B}}$. At $\tau_{\min} = 0$ the initialization returns $\boldsymbol{z}_{\mathbf{0}}$ regardless of $\boldsymbol{x}_{\mathcal{B}}$, so by (a) the output has law $\pi_{\mathcal{B}}$ for every input. At $\tau_{\min} = 1$, the initialization is the identity and the guidance stage is skipped. □

Corollary 1 (Validity of the sweep)**.** *Apply the updates of proposition 1 to a sequence of blocks, each acting on its interior given the current values of all other states. The composition leaves $p(\boldsymbol{x}_{0:N} \mid \boldsymbol{y}_{1:N})$ invariant. If every state is interior to some block (which requires the two edge blocks), $\tau_{\min} < 1$ and $\boldsymbol{\varepsilon} > \mathbf{0}$ on the interior, then under standard regularity conditions the chain is irreducible and aperiodic, so this is its unique stationary distribution (Tierney, 1994; Meyn & Tweedie, 2012).*

Proof. If $\boldsymbol{x}_{0:N} \sim p(\cdot \mid \boldsymbol{y}_{1:N})$ then, conditionally on the states outside a block, its interior is distributed as $\pi_{\mathcal{B}}$ by eq. (8). proposition 1 shows the update preserves this conditional law while leaving the other states unchanged, so the joint law is preserved, and a composition of invariant kernels is invariant. For $\tau_{\min} < 1$ the initialization is a Gaussian kernel with variance $\alpha_{\tau_{\min}}^2 > 0$ and for $\boldsymbol{\varepsilon} > \mathbf{0}$ the guidance stage is an SDE with non-degenerate noise, so both have strictly positive transition densities on the interior. Since every state is interior to some block, one sweep has a strictly positive density on the whole state space, which gives irreducibility and aperiodicity. □

Related results. Proposition 1 is an instance of the observation, made in concurrent work by Kang et al. (2026) and Hill (2026), that a Gaussian noising step followed by exact denoising leaves the data distribution invariant; Gibbs samplers alternating noising and guided denoising with diffusion priors also appear in Coeurdoux et al. (2024) and Wu et al. (2024). The blocked structure of corollary 1 follows the block Gibbs samplers for state-space models of Carter & Kohn (1994), Shephard & Pitt (1997) and Singh et al. (2017).

Relation to the inversion stage. The block smoother initializes the interior directly at $\boldsymbol{\tau}_{\min}$ through eq. (9) rather than inverting it from $\mathbf{1}$. The following proposition shows that this initialization is a special case of the windowed inversion, recovered by choosing a specific diffusion coefficient. The underlying fact, that the generative SDE with this coefficient reduces to the noising step of SDEdit (Meng et al., 2022), is not new; a similar result is Proposition H.1 of Potaptchik et al. (2026).

Proposition 2 (Score-free inversion)**.** *Let $\boldsymbol{\tau}_t$ be a path with $\boldsymbol{\tau}_0 = \mathbf{1}$, $\dot{\boldsymbol{\tau}}_t \leq \mathbf{0}$ componentwise and $\boldsymbol{\tau}_t > \mathbf{0}$, and assume $\beta'\alpha - \alpha'\beta \geq 0$. Let $\boldsymbol{\varepsilon}_{\boldsymbol{\tau}_t} = -\boldsymbol{\lambda}_{\boldsymbol{\tau}_t}$ with $\boldsymbol{\lambda}_{\boldsymbol{\tau}_t}$ as in eq.* (17)*. Then $\boldsymbol{\varepsilon}_{\boldsymbol{\tau}_t} \geq \mathbf{0}$, the SDEs eq.* (4) *and eq.* (6) *coincide along $\boldsymbol{\tau}_t$, and the resulting SDE involves neither $\boldsymbol{\eta}_{\mathbf{0}}, \boldsymbol{\eta}_{\mathbf{1}}$ nor the*

likelihood. Moreover, its solution started from $\boldsymbol{z}$ at $t = 0$ has, for every t, the law of the interpolant eq. (10) *with $\boldsymbol{z_1} = \boldsymbol{z}$,*

$$\boldsymbol{z}_{\boldsymbol{\tau}_t} \overset{d}{=} \alpha_{\boldsymbol{\tau}_t} \odot \boldsymbol{z_0} + \beta_{\boldsymbol{\tau}_t} \odot \boldsymbol{z}, \qquad \boldsymbol{z_0} \sim \rho_{\mathbf{0}} \text{ independent of } \boldsymbol{z}, \tag{24}$$

which extends by continuity to $\boldsymbol{\tau}_t = \mathbf{0}$, where it equals $\boldsymbol{z_0}$.

Proof. Since the drift, score and diffusion coefficient act elementwise and the components of $\boldsymbol{z_0}$ are independent, it suffices to treat a single component. We write τ_t for its flow time, drop the argument τ_t from $\alpha, \beta, \eta_{\mathbf{0}}, \eta_{\mathbf{1}}, \lambda, \varepsilon$, and let $\ell(z) := \partial_z \log p(\boldsymbol{y} \mid z_{\tau_t} = z)$ denote the likelihood score.

By eq. (17) and (A1), $\varepsilon = -\lambda = -\dot\tau_t\, \alpha(\beta'\alpha - \alpha'\beta)/\beta \geq 0$, since $\dot\tau_t \leq 0$, $\alpha \geq 0$, $\beta > 0$ and $\beta'\alpha - \alpha'\beta \geq 0$. Taking $\mathbb{E}[\cdot \mid z_{\tau_t} = z]$ in eq. (10) gives $z = \alpha\, \eta_{\mathbf{0}} + \beta\, \eta_{\mathbf{1}}$, and substituting $\eta_{\mathbf{1}} = (z - \alpha\eta_{\mathbf{0}})/\beta$ into eq. (12)–eq. (13), the drift of eq. (4) becomes

$$b + \varepsilon\, s = \dot\tau_t \frac{\beta'}{\beta}\, z - \frac{\eta_{\mathbf{0}}}{\alpha}\Big(\varepsilon + \dot\tau_t \frac{\alpha(\beta'\alpha - \alpha'\beta)}{\beta}\Big) = \dot\tau_t \frac{\beta'}{\beta}\, z - \frac{\eta_{\mathbf{0}}}{\alpha}\big(\varepsilon + \lambda\big), \tag{25}$$

where the last step is the definition eq. (17). By eq. (15), the guided SDE adds $(\lambda + \varepsilon)\, \ell$ to this drift. At $\varepsilon = -\lambda$ both the $\eta_{\mathbf{0}}$ term and the likelihood term vanish, so the two SDEs reduce to the same linear SDE

$$\mathrm{d}z_{\tau_t} = \dot\tau_t \frac{\beta'}{\beta}\, z_{\tau_t}\, \mathrm{d}t + \sqrt{-2\lambda}\, \mathrm{d}W_t, \tag{26}$$

with continuous coefficients since $\beta > 0$.

Being linear with deterministic coefficients, the solution of eq. (26) from a fixed z is Gaussian, so it suffices to match its mean m_t and variance v_t with eq. (24). These satisfy

$$\dot m_t = \dot\tau_t \frac{\beta'}{\beta}\, m_t, \qquad \dot v_t = 2\dot\tau_t \frac{\beta'}{\beta}\, v_t - 2\dot\tau_t \frac{\alpha(\beta'\alpha - \alpha'\beta)}{\beta}, \qquad m_0 = z,\ v_0 = 0. \tag{27}$$

By the chain rule, $m_t = \beta_{\tau_t} z$ and $v_t = \alpha_{\tau_t}^2$ satisfy $\dot m_t = \dot\tau_t \beta' z = \dot\tau_t \frac{\beta'}{\beta} m_t$ and $\dot v_t = 2\dot\tau_t \alpha\alpha' = 2\dot\tau_t \frac{\beta'}{\beta}\alpha^2 - 2\dot\tau_t \frac{\alpha(\beta'\alpha - \alpha'\beta)}{\beta}$, with $\beta_1 = 1$ and $\alpha_1^2 = 0$; they are therefore the solutions by uniqueness. Hence $z_{\tau_t} \mid z \sim \mathcal{N}(\beta_{\tau_t} z, \alpha_{\tau_t}^2)$, which is the componentwise form of eq. (24). □

D Experiment Details

This appendix gives what is needed to reproduce the results in this paper. We present the DAWIS hyperparameters and how they were tuned, the backbone and training setup, the configuration of every baseline, and the data, splits and observation operators for SQG and `SEVIR`. DAWIS is tuned only on the filtering task, and the same settings are reused unchanged for both smoothers, so the smoothing results involve no additional tuning.

D.1 DAWIS Hyperparameters and tuning

We tune the hyperparameters only for the filtering experiments and use the same settings for the corresponding smoothing experiments. The complete set of hyperparameters is listed in table 3.

Table 3: Hyperparameter configurations for DAWIS for all SQG and `SEVIR` experiments.

Hyperparameter	`Noisy`	`Sparse`	`Multimodal`	`Saturating`	`SEVIR`
Guide Method	MMPS	MMPS	MMPS	MMPS	MMPS
Solver Steps	100	100	100	100	100
ε	0.03	0.03	0.03	0.03	2.5/1.0
$\tau_{\min}^{\text{start}}$	0.4	0.4	0.4	0.4	0.4
$\tau_{\min}^{\text{end}}$	0.3	0.1	0.2	0.4	0.1
$\tau_{\min}^{\text{init}}$	0.0	0.0	0.0	0.0	1.0
$\tau_{\min}^{\text{init-IC}}$	0.9	0.9	0.9	0.9	0.9
Guidance Strength ζ	1	1	1	20	2.5
Guide all steps	False	True	False	True	True

Block smoothers: The Block Gibbs Smoother, DAWIS Block Smoother and DAWIS Soft Block Smoother share the DAWIS window model and the same sweep. Each is initialized from the DAWIS Lagged Smoother on SQG and the DAWIS-Joint Lagged Smoother on `SEVIR`. Blocks of $w+1=7$ states slide over the trajectory with stride 3 and alternate between forward and backward sweeps, with 2 sweeps on SQG and 4 on `SEVIR`. For SQG, the two boundary states of each block condition the update. For a non-Markovian system such as SEVIR, the assumptions of eq. (1) break and the block smoother should condition on more than two points. We tried conditioning on two points for both DAWIS block smoother and Block Gibbs Smoother, but did not see an improvement.

The noising step is a sample from the interpolant eq. (9) followed by 100 guided solver steps with MMPS and $\varepsilon = 0.03$. The three methods differ only in their turning points (table 4). The Block Gibbs Smoother inverts the interior to pure noise ($\tau_{\min} = 0$). The DAWIS Block Smoother uses a tuned interior turning point and keeps the boundary fixed. The soft variant also inverts the boundary, to $\tau_{\min}^{\partial\mathcal{B}} = 0.9$. The guidance strength is $\zeta = 1$ except on `Saturating`, where $\zeta = 150$.

Table 4: Interior turning point $\tau_{\min}$ of the block smoothers. The boundary turning point is 1 for the Block Gibbs and DAWIS Block Smoothers and 0.9 for the DAWIS Soft Block Smoother.

Method	`Noisy`	`Sparse`	`Multimodal`	`Saturating`	`SEVIR`
Block Gibbs Smoother	0	0	0	0	0
DAWIS Block Smoother	0.9	0.7	0.7	0.9	0.9
DAWIS Soft Block Smoother	0.9	0.7	0.7	0.9	0.9

D.2 Backbone Architecture

We parameterize $\eta_{0,1}$ using a U-Net backbone (Ronneberger et al., 2015), with separate output channels for the two components. Preliminary experiments found this to perform comparably to using separate networks for η_0 and η_1 or layer-norm conditioning, while requiring less computation and memory. The backbone follows the general architecture used in Song et al. (2021) and Karras et al. (2022), with the addition of circular padding to respect the periodic boundary conditions of the SQG data. It has three hierarchical levels, with attention at the second level for SQG and only at the deepest level for `SEVIR` due to its higher spatial resolution. Since the model processes a full window of W states jointly, we scale its capacity with the window size by setting the hidden dimension to $32 \times W$ and reduce the batch size accordingly to keep memory usage approximately constant. Unless stated otherwise, all window-based methods use $w + 1 = 7$ states, so the DAWIS Lagged Smoother has a lag of $\ell = 6$ steps (18 h on SQG, 60 min on `SEVIR`). We do not investigate more specialized architectures, but DAWIS is compatible with any model that can be trained using a flow-matching objective.

All DAWIS models use the same backbone architecture and are implemented in PyTorch. Models are trained for 50 epochs on 8 A100 GPUs, using the same optimizer settings across datasets, as listed in table 5.

Table 5: Optimizer hyperparameters, shared across all datasets and window sizes.

Optimizer hyperparameters	
Optimiser	AdamW (Loshchilov & Hutter, 2017a)
Initialization	Xavier Uniform (Glorot & Bengio, 2010)
LR decay schedule	Cosine (Loshchilov & Hutter, 2017b)
Peak LR	1e-3
Minimum LR	1e-5
Weight decay	1e-2
Epochs	50

D.3 BASELINES DETAILS

Here we provide additional details on the baseline methods, including their implementation and hyperparameter settings. Where a baseline was already tuned for this setup in prior work, we reuse those settings. The SDA-based baselines share the DAWIS backbone and are tuned per experiment, so the comparison with them separates the effect of the inference procedure from that of the learned prior.

LETKF: We use the hyperparameters listed in table 6, following the values used in (Andrae et al., 2026), which were already tuned for the experimental settings considered in this paper.

Table 6: Hyperparameter configurations for LETKF.

Hyperparameter	`Noisy`	`Sparse`	`Multimodal`	`Saturating`	`SEVIR`
Localization scale	2500 km	1500 km	1500 km	4500 km	3 px
Relaxation to prior spread	0.5	0.5	0.5	0.9	0.0

FlowDAS: For `SEVIR`, we use the pretrained backbone and original hyperparameters from Chen et al. (2025). For SQG, we use the backbone and hyperparameters from Andrae et al. (2026), which were tuned for this experimental setting. All hyperparameters for all experiments for FlowDAS are shown in table 7.

Table 7: Hyperparameter configurations for FlowDAS for all SQG and `SEVIR` experiments.

Experiment	**Solver Steps**	**MC Samples**	**Window**	**Guidance Strength ζ**
`Noisy`	500	25	6	1
`Sparse`	500	25	6	3
`Multimodal`	500	25	6	0.5
`Saturating`	500	25	6	5
`SEVIR`	500	25	6	0.1

EnSF: We use the same implementation and hyperparameters as in Andrae et al. (2026) shown in table 8.

Table 8: Hyperparameter configurations for EnSF for all SQG and `SEVIR` experiments.

Experiment	**Solver Steps**	ϵ_α
`Noisy`	1000	0.05
`Sparse`	1000	0.05
`Multimodal`	1000	0.05
`Saturating`	1000	0.05
`SEVIR`	1000	0.05

SDA Filter: We use the same backbone that we use for DAWIS and we tune the hyperparameters for each experiment. The final hyperparameters are shown in table 9.

Table 9: Hyperparameter configurations for SDA Filter for all SQG and SEVIR experiments.

Experiment	Guide Method	Solver Steps	ε	Corrections	τ	Guidance Strength ζ
Noisy	MMPS	100	1	1	0.3	2
Sparse	MMPS	100	1	1	0.3	2
Multimodal	MMPS	100	1	1	0.5	2
Saturating	MMPS	100	1	1	0.3	20
SEVIR	MMPS	100	1	1	0.3	10

DAISI: We use the same hyperparameters (see table 10) and pretrained backbone as in the original DAISI implementation (Andrae et al., 2026).

Table 10: Hyperparameter configurations for DAISI for all SQG and SEVIR experiments.

Experiment	Guide Method	Solver Steps	ε	$t_{\min}$	Guidance Strength ζ
Noisy	DPS	100	0.03	0.4	10
Sparse	MMPS	100	0.03	0.3	1
Multimodal	MMPS	100	0.03	0.3	1
Saturating	MMPS	100	0.03	0.3	20
SEVIR	MMPS	100	0.03	0.3	1

Joint AR: Both variants use the same window model as DAWIS and no separate forecast model. Each cycle generates part of the window from pure noise, conditioned on the rest of the window held at data ($\tau = 1$) (Shysheya et al., 2024). Joint AR 1|W holds the $w = 6$ most recent analyses fixed and generates only the new state, guided by its observation. Joint AR W|1 holds only the oldest state fixed and regenerates the other six, guided by every observation in the window, so it also revises past states. Neither variant has an inversion step. The hyperparameters, tuned per experiment, are shown in table 11.

Table 11: Hyperparameter configurations for Joint AR for all SQG and SEVIR experiments. Both variants use MMPS and $\varepsilon = 1$.

	Joint AR 1|W		Joint AR W|1	
Experiment	Solver Steps	ζ	Solver Steps	ζ
Noisy	100	5	100	4
Sparse	100	5	100	7
Multimodal	100	5	25	5
Saturating	100	20	100	150
SEVIR	100	3	100	1

ForcingDAS-Pyr: We implement the pyramid schedule of Jia et al. (2026) on the DAWIS window model, so both methods share the same prior. The window of $T = w + 1 = 7$ states sits on a noise ladder: at the start of each cycle, state j is at flow time $\tau^j = (T - 1 - j)/(T - 1)$ for $j \geq 1$. Each cycle advances every state one rung of $1/(T - 1)$ under guidance from all observations in the window. The first state is held at data as clean context ($n_{\text{fixed}} = 1$), so each state is finalized after $w - 1 = 5$ future observations. All experiments use MMPS, 20 solver steps per cycle and $\varepsilon = 0.03$. The guidance strength is tuned per experiment: $\zeta = 2, 15, 4, 150$ and 3 for Noisy, Sparse, Multimodal, Saturating and SEVIR.

SDA: We use the same backbone that we use for DAWIS and we tune the hyperparameters for each experiment. The final hyperparameters are shown in table 12.

Table 12: Hyperparameter configurations for SDA for all SQG and SEVIR experiments.

Experiment	Guide Method	Solver Steps	ε	Corrections	τ	Guidance Strength ζ
Noisy	MMPS	100	1	1	0.5	2
Sparse	MMPS	100	1	1	0.5	5
Multimodal	MMPS	200	1	3	1.0	3
Saturating	MMPS	100	1	2	0.5	30
SEVIR	MMPS	100	1	1	0.5	3

D.4 Surface Quasi-Geostrophic (SQG)

For the SQG experiments, we reuse the training dataset from Andrae et al. (2026), consisting of 2,000 trajectories of 100 states sampled at 3-hour intervals. Each trajectory is initialized from a random state and spun up for 300 days to reach approximate stationarity before the recorded trajectory begins. Evaluation is performed on 10 held-out trajectories of the same length, with metrics averaged over the final 20 states. Since the data are approximately mean-zero, we normalize each channel only by its standard deviation. The experimental settings are summarized in table 13, and the corresponding observation operators are visualized in fig. 3.

Table 13: Experiment configurations.

Experiment	Data	Obs operator	σ_{obs}	Sparsity
Noisy	SQG-64	x	5	25%
Sparse	SQG-64	x	1	5%
Multimodal	SQG-64	$(x/7)^2$	1	25%
Saturating	SQG-64	$\arctan(x)$	0.01	25%
SEVIR	SEVIR	x	0.001	10%

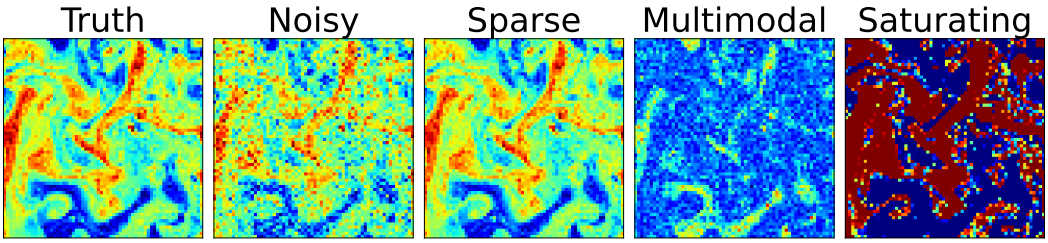


Figure 3: Visualization of the observations for each configuration before sparsity is applied.

D.5 SEVIR

Data and split. We use the SEVIR VIL low-resolution product (128×128, 25 frames per trajectory at 10 min intervals). Following Chen et al. (2025), events after 2019-06-01 form the test set; earlier events are split into training and validation by a seeded random 10 % holdout. This gives 13,434 / 1,492 / 4,053 train / validation / test events. We report results on 10 test trajectories chosen randomly from the test set.

Normalization. All networks trained here use the FlowDAS normalization $(\text{VIL}/255 - 0.5)/0.1$ except for DAISI's pretrained prior which was trained on $\text{VIL}/255$. All metrics are computed in raw VIL units (0–255), so scores are comparable across methods whose networks use different normalizations.

Initialization. The first 6 frames of each trajectory are given as initialization without any noise following Chen et al. (2025), and the remaining 19 frames are assimilated. Filters are scored over the last 10 steps and smoothers over all 19.

E ADDITIONAL RESULTS

This appendix extends the CRPS evaluation in table 2 with the RMSE (table 14) and the spread-skill ratio (SSR, table 15), and breaks the scores down by setting, over time, by calibration and by spectral content. The RMSE ranking closely follows the CRPS ranking. On SQG, DAWIS Filter, DAWIS Lagged Smoother and DAWIS Block Smoother are the best, or tied for best, in their category in every setting. On `SEVIR`, the DAWIS Block Smoother has the lowest RMSE overall. The SSR shows that the DAWIS filters stay close to calibration on SQG. SDA-Filter and EnSF are clearly underdispersive in most settings. The DAWIS lagged smoothers are mildly overdispersive on `Saturating`. On `SEVIR`, all DAWIS variants are overdispersive and the SSR varies widely between trajectories.

Table 14: The RMSE for experiments on SQG and `SEVIR`. We display the mean and standard deviation across 10 independent trajectories, averaged over the last 20 (10 for `SEVIR`) steps. The best score for each experiment is highlighted in **bold** and the second best with an underline. Smoothing methods calculate the metrics over the whole trajectory.

Method	Noisy	Sparse	Multimodal	Saturating	SEVIR
Filtering					
EnSF Bao et al. (2024)	7.80±1.07	6.95±0.82	3.31±1.95	8.80±2.39	35.12±26.68
FlowDAS Chen et al. (2025)	4.87±0.79	6.09±0.94	4.84±0.81	6.77±0.61	27.31±13.16
LETKF Liang et al. (2025)	2.49±0.26	4.31±0.47	8.97±2.04	4.19±1.86	12.19±7.04
DAISI Andrae et al. (2026)	2.42±0.25	3.31±0.40	2.99±0.19	3.65±0.99	11.02±6.10
SDA-Filter	2.36±0.22	3.16±0.40	3.69±0.57	5.98±0.77	10.05±5.00
Joint AR 1|W Shysheya et al. (2024)	6.89±0.67	7.37±0.90	7.22±0.70	6.86±0.78	16.19±7.84
Joint AR W|1 Shysheya et al. (2024)	2.28±0.22	3.23±0.31	2.22±0.16	2.23±0.27	11.10±6.50
DAWIS Filter	**1.91**±0.18	**2.40**±0.28	**1.96**±0.15	**1.71**±0.23	10.14±5.08
DAWIS-Joint Filter	2.20±0.23	2.47±0.30	2.29±0.24	1.93±0.29	**9.97**±5.42
Fixed-lag Smoothing					
SDA-Filter	2.43±0.13	2.73±0.21	3.14±0.42	4.91±0.60	10.94±5.83
ForcingDAS-Pyr Jia et al. (2026)	2.55±0.14	2.53±0.21	2.03±0.17	2.12±0.27	9.52±5.94
Joint AR W|1 Shysheya et al. (2024)	2.01±0.09	2.53±0.10	1.94±0.08	1.98±0.15	9.73±6.28
DAWIS Lagged Smoother	**1.72**±0.08	**2.02**±0.11	**1.75**±0.07	**1.48**±0.10	**9.05**±5.24
DAWIS-Joint Lagged Smoother	1.98±0.11	2.08±0.12	2.04±0.11	1.62±0.13	9.09±5.31
Joint Smoothing					
SDA Rozet & Louppe (2023)	1.92±0.12	1.94±0.11	5.63±0.59	2.11±0.10	9.89±5.78
Block Gibbs Smoother	1.70±0.10	**1.92**±0.12	**1.68**±0.09	1.66±0.15	9.27±5.63
DAWIS Block Smoother	**1.67**±0.09	**1.92**±0.12	**1.68**±0.08	**1.36**±0.11	**8.93**±5.45

Table 15: The SSR for experiments on SQG and SEVIR. We display the mean and standard deviation across 10 independent trajectories, averaged over the last 20 (10 for SEVIR) steps. The best score for each experiment is highlighted in **bold** and the second best with an underline. Smoothing methods calculate the metrics over the whole trajectory.

Method	Noisy	Sparse	Multimodal	Saturating	SEVIR
Filtering					
EnSF Bao et al. (2024)	0.43±0.05	0.49±0.05	1.24±0.33	0.48±0.39	0.35±0.21
FlowDAS Chen et al. (2025)	0.66±0.07	0.96±0.06	0.87±0.11	**1.00**±0.09	0.53±0.23
LETKF Liang et al. (2025)	**1.00**±0.05	1.05±0.07	0.78±0.07	0.63±0.12	1.16±0.40
DAISI Andrae et al. (2026)	0.89±0.03	1.04±0.04	1.04±0.19	1.34±0.18	0.92±0.19
SDA-Filter	0.72±0.02	0.60±0.06	0.64±0.06	0.43±0.07	1.34±0.82
Joint AR 1|W Shysheya et al. (2024)	1.09±0.07	1.09±0.11	1.03±0.07	1.28±0.10	0.92±0.08
Joint AR W|1 Shysheya et al. (2024)	0.74±0.04	**0.97**±0.03	0.86±0.04	0.66±0.04	**0.95**±0.34
DAWIS Filter	0.89±0.01	1.09±0.06	0.96±0.01	1.11±0.04	1.29±0.81
DAWIS-Joint Filter	0.92±0.03	1.07±0.05	**0.98**±0.03	1.07±0.03	1.36±0.89
Fixed-lag Smoothing					
SDA-Filter	0.74±0.01	0.61±0.05	0.64±0.06	0.51±0.08	1.16±0.63
ForcingDAS-Pyr Jia et al. (2026)	0.90±0.02	1.02±0.02	**0.97**±0.03	**1.02**±0.02	**0.97**±0.16
Joint AR W|1 Shysheya et al. (2024)	0.77±0.04	**0.99**±0.02	0.89±0.04	0.71±0.02	1.06±0.50
DAWIS Lagged Smoother	0.99±0.01	1.15±0.03	1.06±0.01	1.21±0.02	1.25±0.48
DAWIS-Joint Lagged Smoother	**1.00**±0.02	1.15±0.03	1.06±0.02	1.16±0.02	1.27±0.50
Joint Smoothing					
SDA Rozet & Louppe (2023)	0.69±0.01	0.73±0.01	0.93±0.07	0.63±0.03	1.48±0.99
Block Gibbs Smoother	0.89±0.01	**1.03**±0.04	0.94±0.01	**1.00**±0.02	**1.16**±0.59
DAWIS Block Smoother	**0.92**±0.01	1.06±0.04	**0.95**±0.01	1.12±0.02	1.31±0.67

E.1 SCORECARDS

Figures 4 and 5 show the relative improvement of each method over a reference. The reference is DAWIS Filter for filtering and DAWIS Lagged Smoother for both smoothing categories, so a negative entry means the method is worse than DAWIS. First, the margins on SQG are large: the strongest filtering baseline, Joint AR W|1, is still 16–39% worse than DAWIS Filter in CRPS. On SEVIR the gaps close, and SDA-Filter and DAWIS-Joint Filter are slightly better than the reference. Second, the joint-smoothing rows show what the block sweep adds on top of its initialization. The DAWIS Block Smoother improves by 3–11%, most on Saturating. The Block Gibbs Smoother, which regenerates from pure noise, makes the initialization worse by 14% on Saturating and 6% on SEVIR.

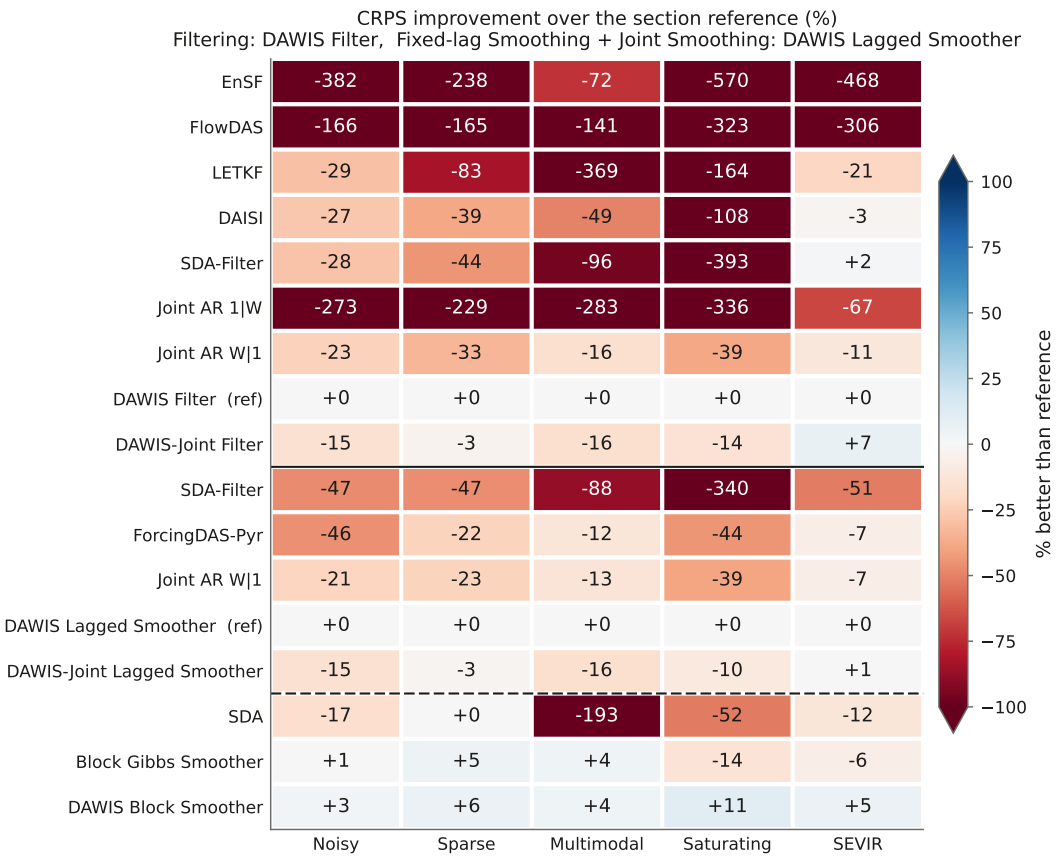


Figure 4: CRPS scorecard against DAWIS.

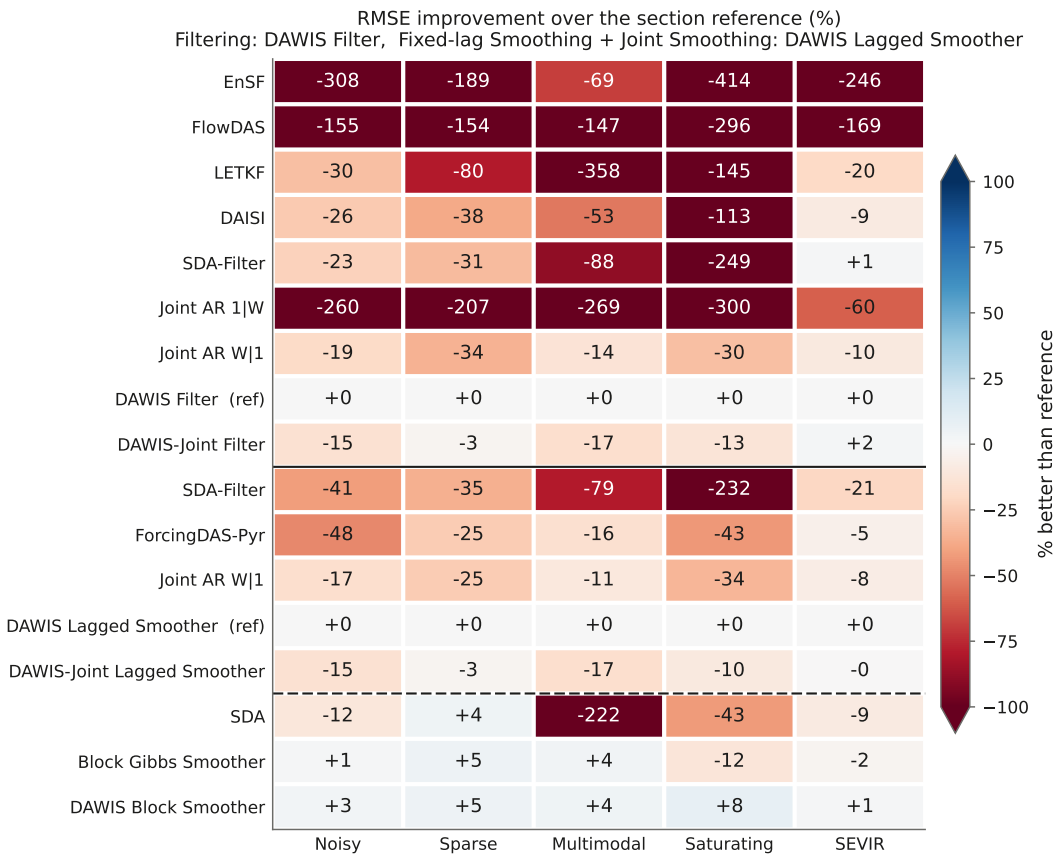


Figure 5: RMSE scorecard against DAWIS.

E.2 Scores over time – filtering

Figures 6 and 7 show the analysis error at every assimilation step, averaged over the 10 trajectories. After a spin-up of roughly 50 h, both DAWIS filters settle at a constant error level and stay there for the rest of the trajectory. This supports averaging over the last 20 steps in table 2. Several baselines instead drift. EnSF, FlowDAS and Joint AR 1|W accumulate error throughout the trajectory in every SQG setting. LETKF diverges within the first ∼50 h on `Multimodal`. On `Saturating`, LETKF and DAISI start to degrade late in the trajectory.

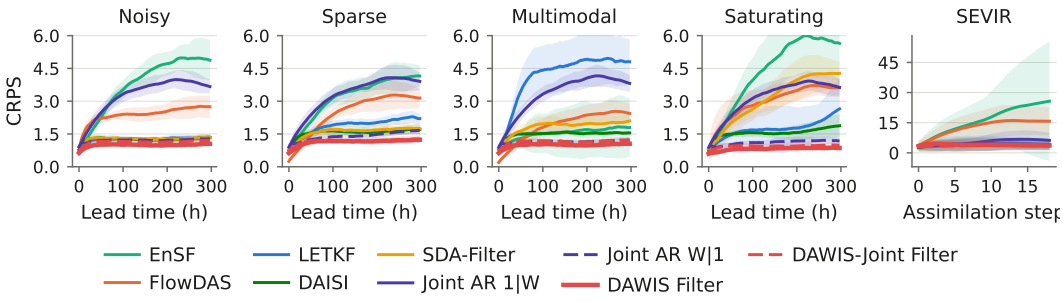


Figure 6: The CRPS over the assimilation window for the filtering methods.

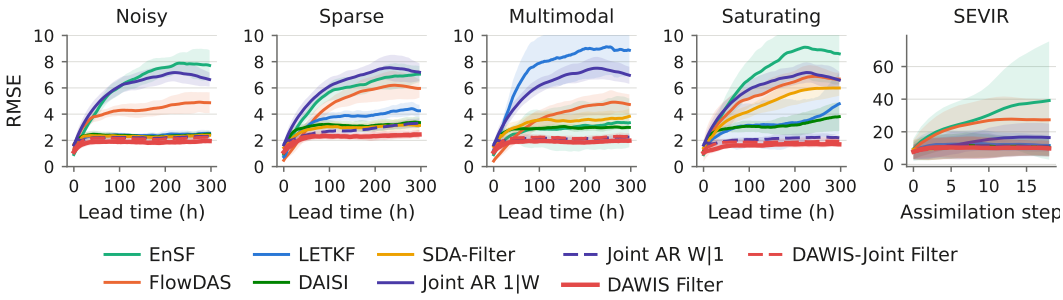


Figure 7: The RMSE over the assimilation window for the filtering methods.

E.3 Scores over time – smoothing

Figures 8 and 9 show the error at every assimilation step, averaged over the 10 trajectories for the smoothers. The DAWIS lagged smoothers have flat error over the whole trajectory. All fixed-lag smoothers show an increase in error over the last few steps, where fewer than ℓ future observations are available and the smoother reduces to a filter. Figure 10 isolates the gain from look-ahead within a single DAWIS run. After spin-up, the lagged estimate is below the filter estimate at every step, and the gap closes at the final step. This gain comes at no extra cost, because both estimates come from the same cycle.

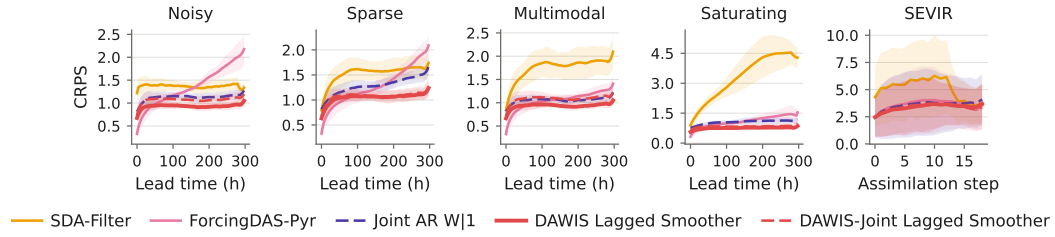


Figure 8: The CRPS over the assimilation window for all smoothing methods.

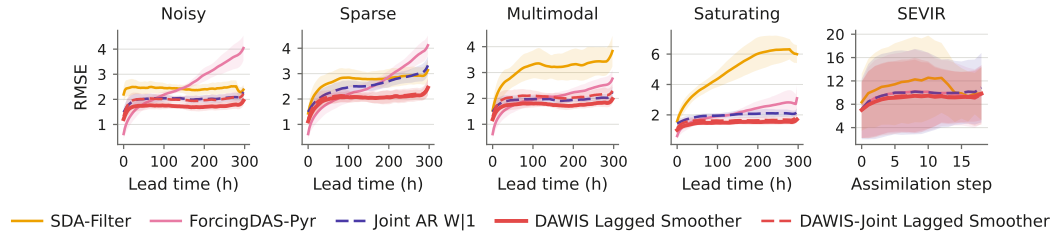


Figure 9: The RMSE over the assimilation window for all smoothing methods.

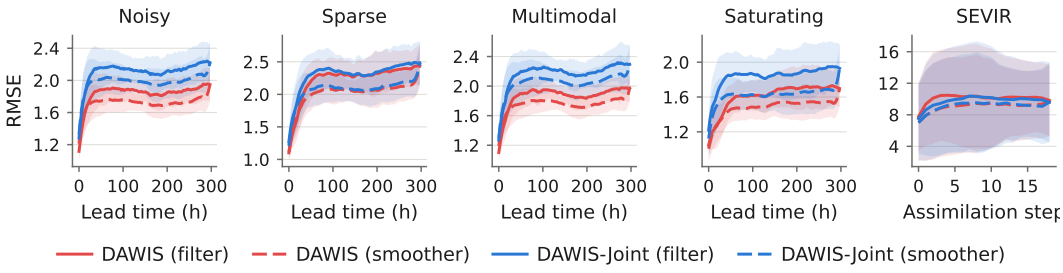


Figure 10: The improvement in RMSE gained by using the lagged smoother for the DAWIS experiments.

E.4 Calibration

We assess calibration with the spread-skill ratio over time (fig. 11) and with verification-rank histograms. A calibrated ensemble has SSR ≈ 1 and a flat histogram. SSR < 1 with a U-shaped histogram indicates underdispersion, and SSR > 1 with a dome-shaped histogram indicates overdispersion. All ensembles start overdispersive. After this transient, DAWIS stays close to one for the whole trajectory in every SQG setting. SDA-Filter and EnSF settle at 0.4–0.7, and DAISI and Joint AR 1|W drift towards overdispersion on `Saturating`. The rank histograms agree. The DAWIS histograms are nearly flat, with only slightly raised end bins. On `Multimodal`, LETKF, FlowDAS, Joint AR W|1 and especially SDA-Filter are U-shaped, and EnSF is dome-shaped.

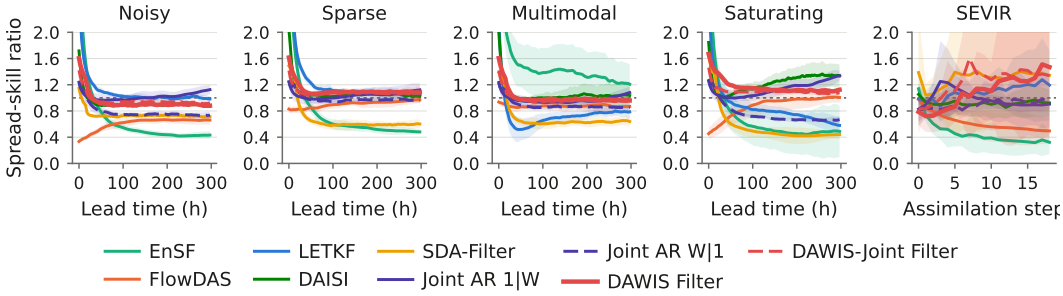


Figure 11: Spread-skill ratio of the filtering methods against its ideal value of one.

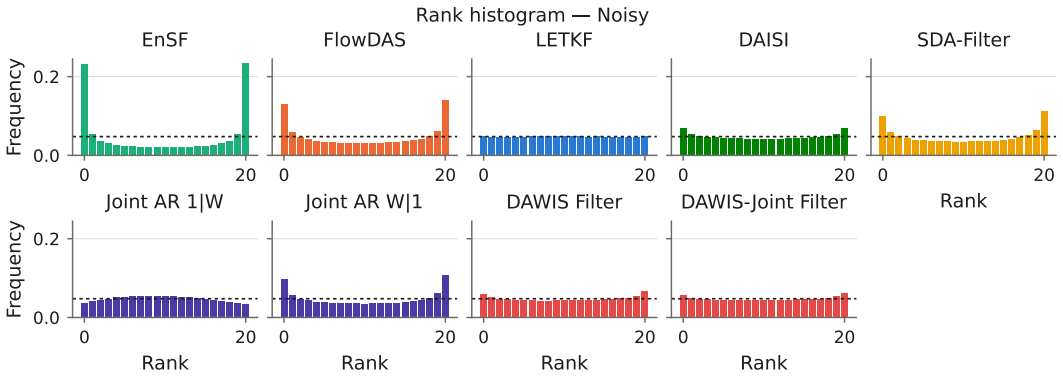


Figure 12: Rank histograms of the filtering methods for `Noisy`.

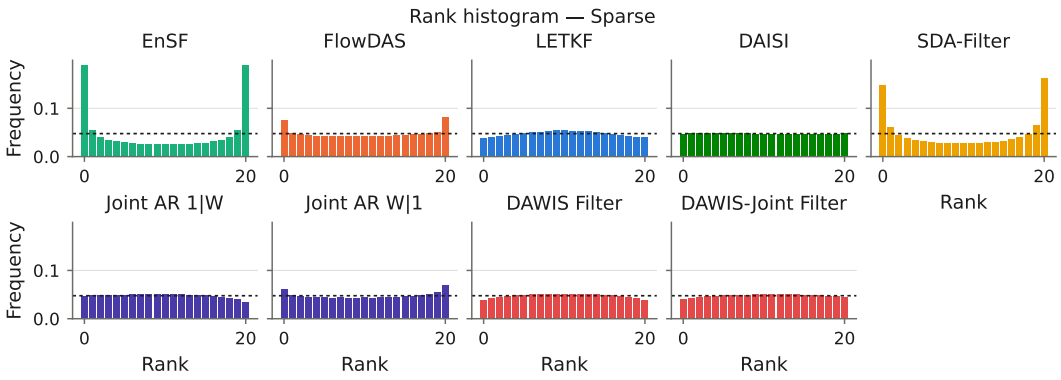


Figure 13: Rank histograms of the filtering methods for `Sparse`.



Figure 14: Rank histograms of the filtering methods for `Multimodal`.

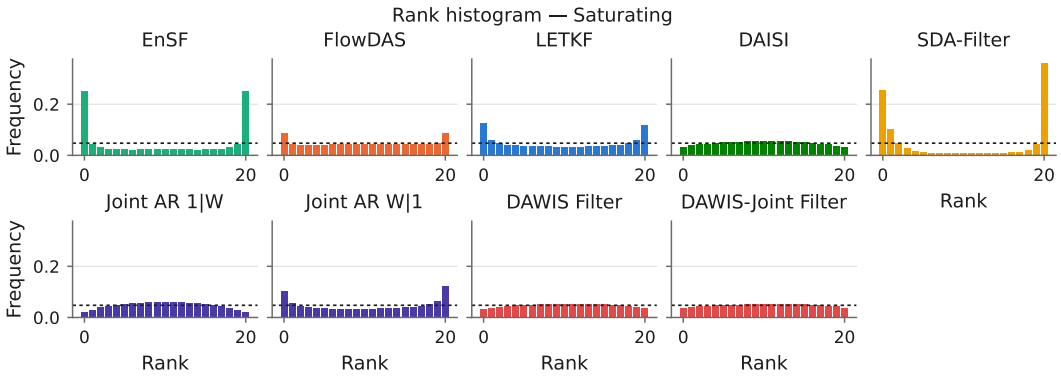


Figure 15: Rank histograms of the filtering methods for `Saturating`.

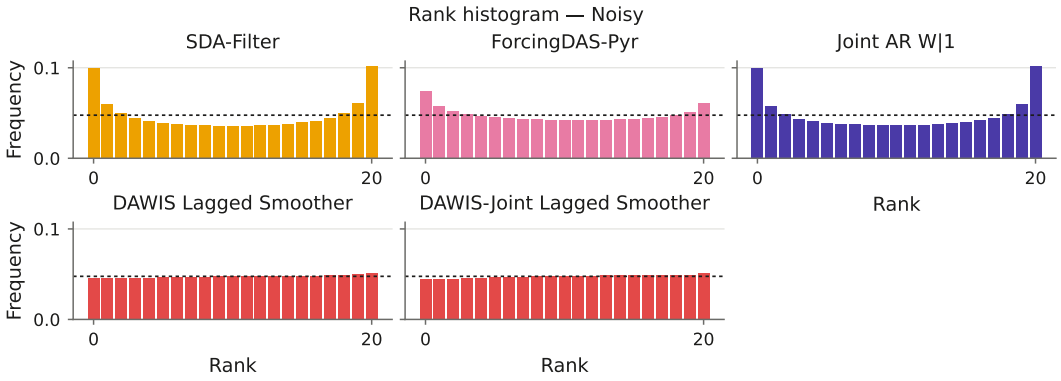


Figure 16: Rank histograms of the smoothing methods for `Noisy`.

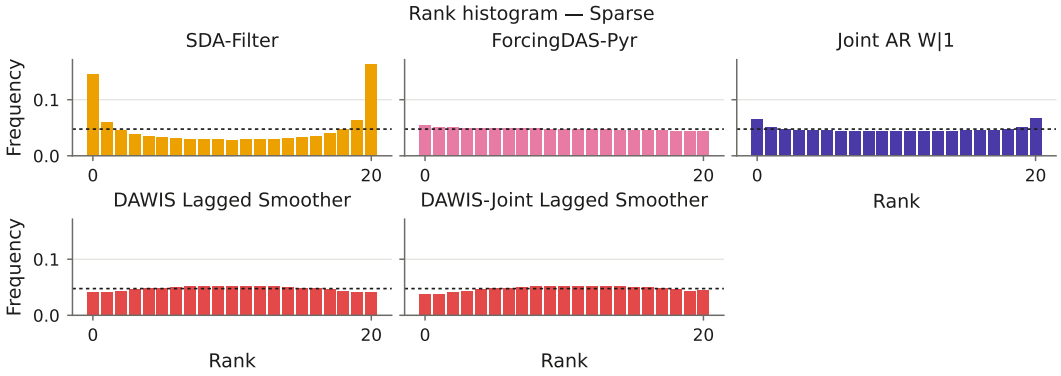


Figure 17: Rank histograms of the smoothing methods for `Sparse`.

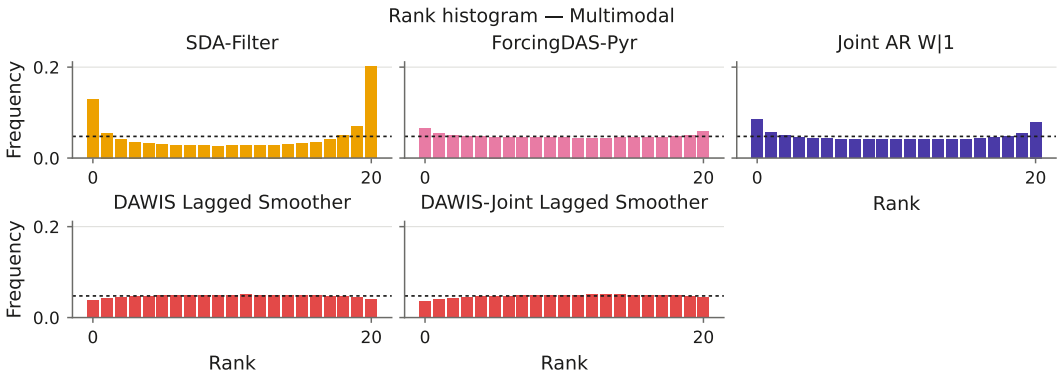


Figure 18: Rank histograms of the smoothing methods for `Multimodal`.

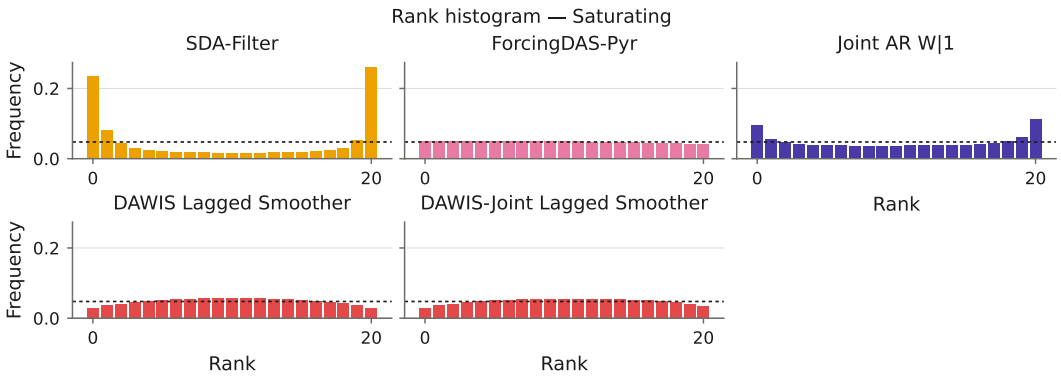


Figure 19: Rank histograms of the smoothing methods for `Saturating`.

E.5 Power spectra

Pointwise scores such as CRPS cannot tell whether a method blurs the fields or adds spurious small-scale noise. The radially averaged power spectra can (figs. 20 and 21), and the log spectral distance (LSD, fig. 22) is used to aggregate and tracks the discrepancy over time. DAWIS matches the true spectrum across the resolved range in every setting. EnSF and FlowDAS have too much power at high wavenumbers on SQG. LETKF has too much power at intermediate and high wavenumbers on `Multimodal`, consistent with the grid-scale noise in fig. 33. SDA-Filter has too little power at all scales on `Saturating`. On `SEVIR`, EnSF underestimates the large scales and overestimates the small ones. The LSD shows that these errors are not transient. The spectral error grows over time for LETKF on `Multimodal`, FlowDAS and Joint AR 1|W on `Sparse`, and SDA-Filter on `Saturating`, while DAWIS stays lowest and constant. After smoothing, every method reproduces the spectrum except SDA-Filter, which lacks power on `Saturating` and has too much at the smallest `SEVIR` scales.

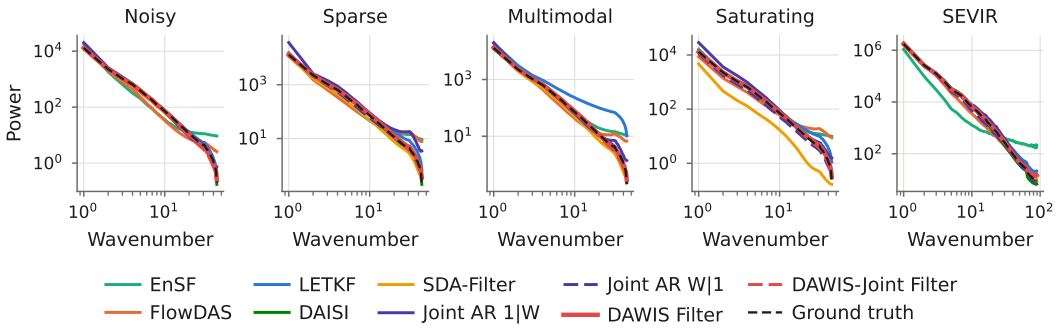


Figure 20: Radially averaged power spectra of the filtering ensemble.

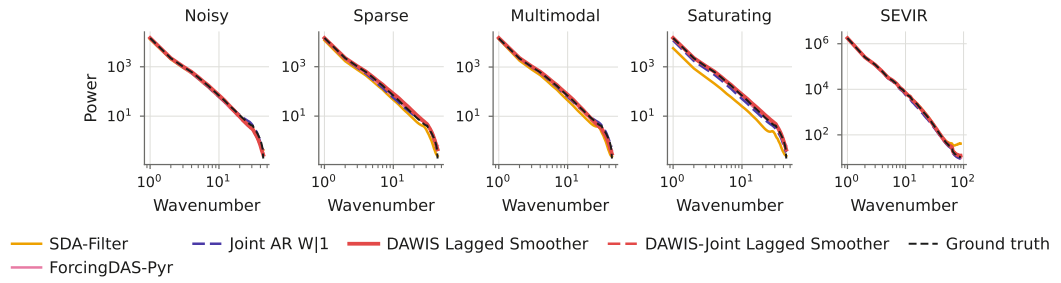


Figure 21: Radially averaged power spectra of the lagged smoothing ensemble.

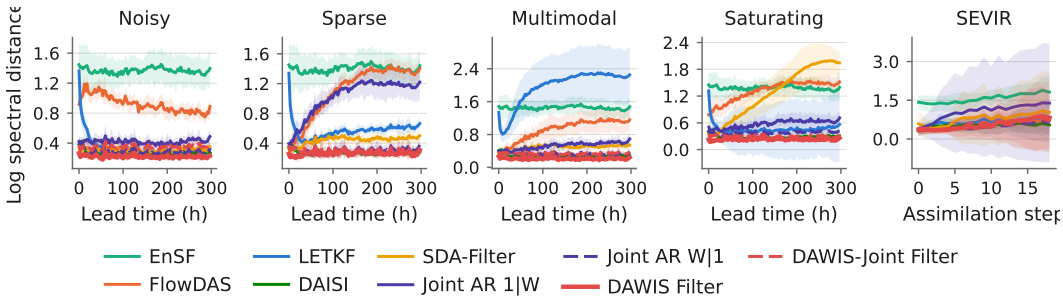


Figure 22: Log spectral distance over the assimilation window for the filtering methods.

F Ablation Experiments

Here we show ablation experiments. In the main SQG experiments, the forecast model is the model that generated the data, so section F.1 tests robustness to a misspecified forward model. Section F.2 relaxes the fixed boundary of the block smoother. In both cases DAWIS degrades gracefully. Under model error, the DAWIS Filter CRPS rises by at most 0.15. DAWIS-Joint does not integrate a forecast model and stays at or below its well-specified CRPS, overtaking DAWIS on `Sparse` and `Saturating`.

F.1 Model error

In the SQG experiments above, the numerical model available to each method at test time is identical to the model used to generate the data. This gives an advantage to methods that explicitly integrate the dynamics, since they have access to the exact model governing the data. To assess robustness to model error, we therefore repeat the experiments with a misspecified numerical model.

The ground-truth trajectories are generated using an SQG simulation at 256×256 resolution and subsequently downsampled to the 64×64 grid used by the assimilating methods. Both the numerical forward model and the learned generative prior operate at 64×64, so the unresolved scales of the high-resolution simulation introduce model error that is not represented by the assimilating model. The observation operators, noise levels, and trajectories are otherwise identical to the corresponding experiments above.

The results in table 16 show that DAWIS consistently outperforms 4D-Var in both CRPS and RMSE across all four observation settings. The DAWIS-Joint variants achieve comparable performance to the standard DAWIS variants while not requiring a separate forecasting model. The difference is particularly pronounced for the smoothing methods, where both DAWIS variants substantially reduce CRPS and RMSE relative to 4D-Var. The SSR results further illustrate the effect of model misspecification on calibration. 4D-Var is substantially underdispersive in the `Noisy` setting and overdispersive in `Sparse` and `Multimodal`, while the DAWIS variants remain closer to the target value of one across the different observation scenarios. Overall, these results indicate that DAWIS maintains accurate and comparatively well-calibrated uncertainty estimates even when the dynamics used during assimilation do not match those generating the data.

F.2 DAWIS Soft Block Smoother (adjust endpoints)

The DAWIS Block Smoother holds the two boundary states fixed, which is what makes each update an exact conditional sampling (proposition 1). A natural relaxation also lets the boundary move, inverting it to a turning point $\tau_{\min}^{\partial\mathcal{B}} < 1$ (we use 0.9) while the interior is inverted to $\tau_{\min}$ as before. The update is then no longer a Gibbs step on $\pi_{\mathcal{B}}$, so we make no invariance claim and instead evaluate empirical performance. Table 17 shows that on SQG the soft variant performs within reported precision of the standard one on `Noisy`, `Sparse` and `Multimodal`, and slightly better on `Saturating` (CRPS 0.65 vs. 0.68). Holding the boundary therefore costs little in practice, while potentially improving empirical performance in certain settings.

Table 16: The CRPS, RMSE and SSR under forward-model error. The truth is a higher-resolution SQG run downsampled onto the 64×64 grid, so the model each method integrates is not the model that generated the data; the observation operator, the noise and the trajectories are those of the corresponding experiment above. We display the mean and standard deviation across 10 independent trajectories, averaged over the last 20 steps for filtering methods and over the whole trajectory for smoothing methods. The best score for each experiment is highlighted in **bold** and the second best with an underline. For SSR, best is closest to 1.

Method	Noisy (ds)	Sparse (ds)	Multimodal (ds)	Saturating (ds)
		CRPS ↓		
Filtering				
DAWIS Filter	**1.12**±0.14	1.22±0.16	**1.13**±0.13	1.01±0.15
DAWIS-Joint Filter	1.13±0.13	**1.19**±0.17	1.18±0.13	**0.95**±0.16
Smoothing				
4D-Var	1.35±0.04	1.33±0.17	1.16±0.04	0.94±0.08
DAWIS Lagged Smoother	**0.98**±0.05	1.06±0.06	**1.02**±0.05	0.83±0.06
DAWIS-Joint Lagged Smoother	1.04±0.05	**1.05**±0.06	1.10±0.06	**0.81**±0.07
		RMSE ↓		
Filtering				
DAWIS Filter	**2.08**±0.26	2.40±0.33	**2.15**±0.24	1.95±0.28
DAWIS-Joint Filter	2.13±0.24	**2.36**±0.35	2.27±0.25	**1.87**±0.31
Smoothing				
4D-Var	2.35±0.08	2.65±0.35	2.17±0.09	1.80±0.15
DAWIS Lagged Smoother	**1.84**±0.09	2.06±0.12	**1.94**±0.09	1.63±0.10
DAWIS-Joint Lagged Smoother	1.96±0.10	**2.04**±0.12	2.10±0.12	**1.59**±0.12
		SSR (→ 1)		
Filtering				
DAWIS Filter	0.79±0.01	**1.06**±0.02	0.87±0.03	0.92±0.01
DAWIS-Joint Filter	**0.93**±0.02	1.09±0.03	**0.98**±0.03	**1.07**±0.03
Smoothing				
4D-Var	0.77±0.01	1.36±0.03	1.15±0.07	**1.00**±0.06
DAWIS Lagged Smoother	0.92±0.01	**1.12**±0.01	**0.98**±0.02	1.04±0.01
DAWIS-Joint Lagged Smoother	**1.02**±0.02	1.17±0.02	1.06±0.03	1.14±0.01

Table 17: The CRPS, RMSE and SSR for experiments on SQG and `SEVIR`. We display the mean and standard deviation across 10 independent trajectories, averaged over the whole trajectory. The best score for each experiment is highlighted in **bold** and the second best with an underline. For SSR, best is closest to 1.

Method	Noisy	Sparse	Multimodal	Saturating	SEVIR
			CRPS ↓		
Block Gibbs Smoother	0.92±0.06	0.99±0.07	0.90±0.05	0.86±0.09	3.58±2.35
DAWIS Block Smoother	0.90±0.05	0.98±0.06	0.89±0.05	0.68±0.06	3.18±2.17
DAWIS Soft Block Smoother	0.91±0.05	0.97±0.06	0.89±0.05	0.65±0.06	3.15±2.19
			RMSE ↓		
Block Gibbs Smoother	1.70±0.10	1.92±0.12	1.68±0.09	1.66±0.15	9.27±5.63
DAWIS Block Smoother	1.67±0.09	1.92±0.12	1.68±0.08	1.36±0.11	8.93±5.45
DAWIS Soft Block Smoother	1.68±0.09	1.90±0.12	1.67±0.08	1.32±0.11	8.92±5.47
			SSR (→ 1)		
Block Gibbs Smoother	0.89±0.01	1.03±0.04	0.94±0.01	**1.00**±0.02	1.16±0.59
DAWIS Block Smoother	0.92±0.01	1.06±0.04	0.95±0.01	1.12±0.02	1.31±0.67
DAWIS Soft Block Smoother	0.90±0.01	1.05±0.03	0.95±0.01	1.11±0.02	1.33±0.72

G STATE FIELDS

We show qualitative examples for each setting. For every method, the panels show the ensemble mean, one member, the ensemble spread and the absolute error of the mean: for filtering at the final assimilation step (figs. 23 to 27), and for smoothing halfway through the trajectory (figs. 28 to 32). The failure modes seen in the aggregate scores show up clearly here. On `Multimodal`, LETKF's analysis turns into grid-scale noise. The Joint AR $1|\mathrm{W}$ members are individually plausible but disagree with each other, so their mean collapses into a featureless band. FlowDAS members have noisy, unphysical patches. Figure 33 follows the `Multimodal` filter mean over time. LETKF loses the flow structure within $48\,\mathrm{h}$. DAISI stays coherent but gradually loses the high-frequency features, drifting towards an overly smooth field. The DAWIS filter tracks the high-frequency features better throughout the trajectory.

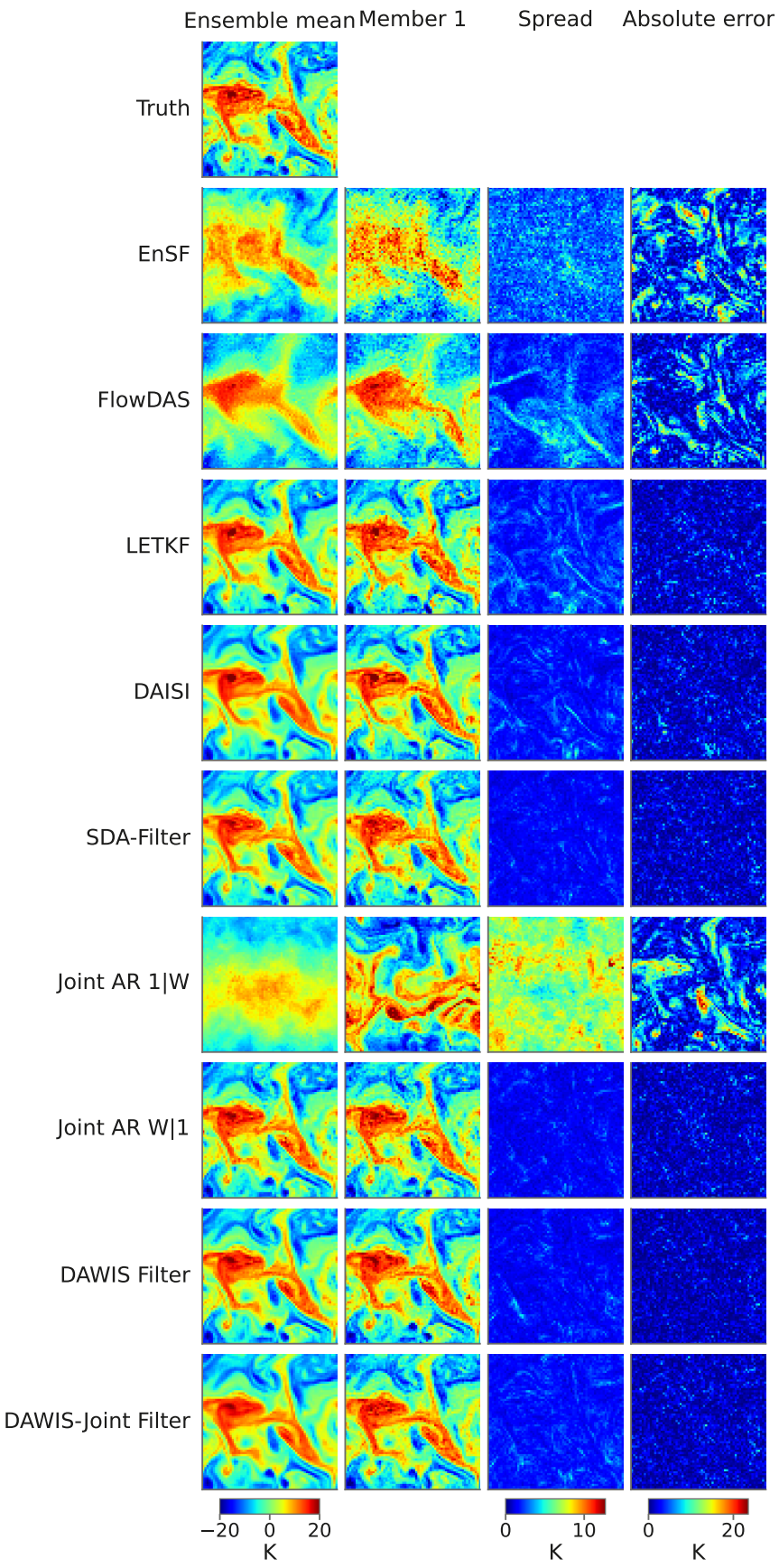


Figure 23: Filtering states at the end of the trajectory for SQG `Noisy`.

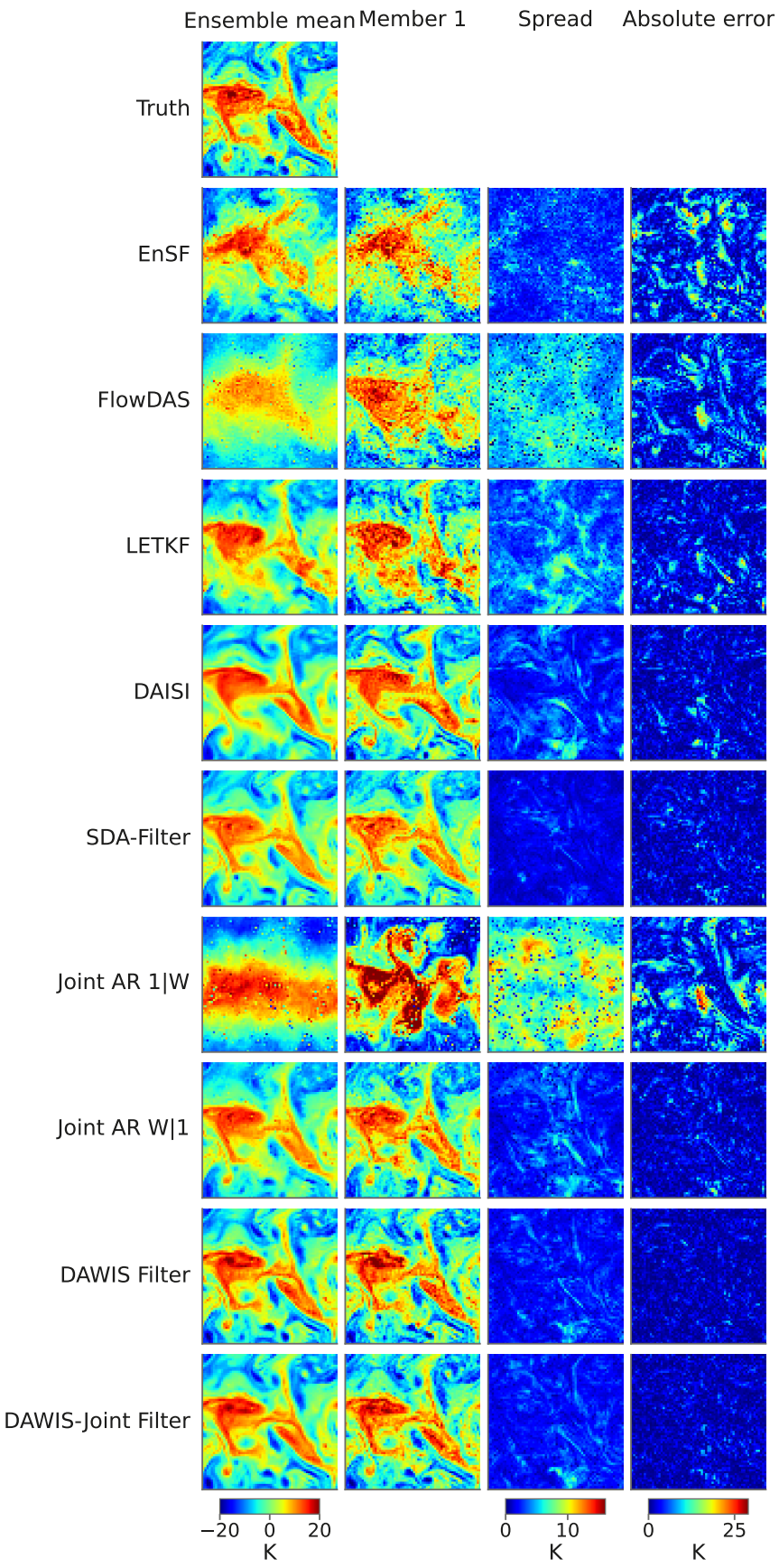


Figure 24: Filtering states at the end of the trajectory for SQG `Sparse`.

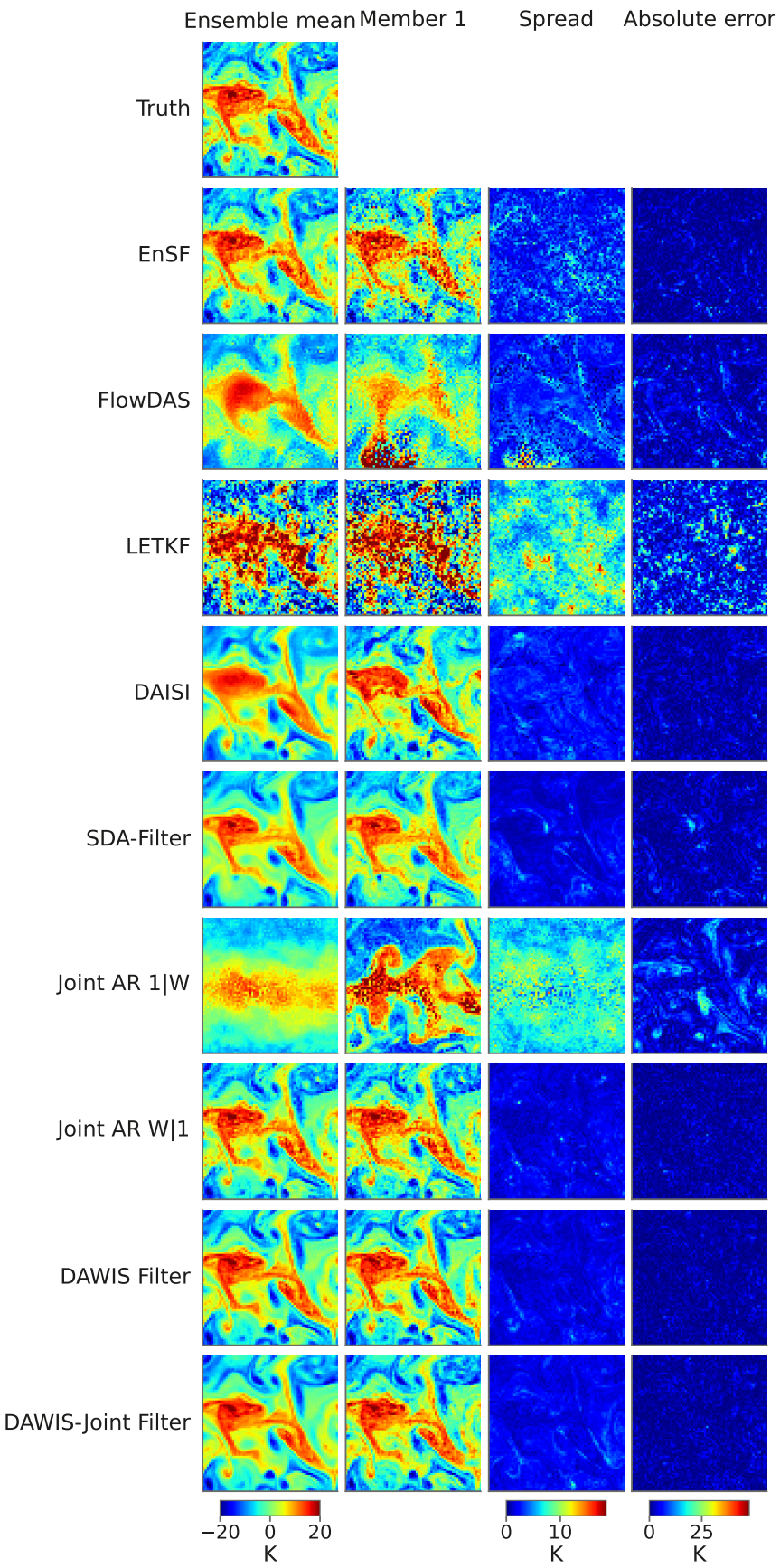


Figure 25: Filtering states at the end of the trajectory for SQG `Multimodal`.

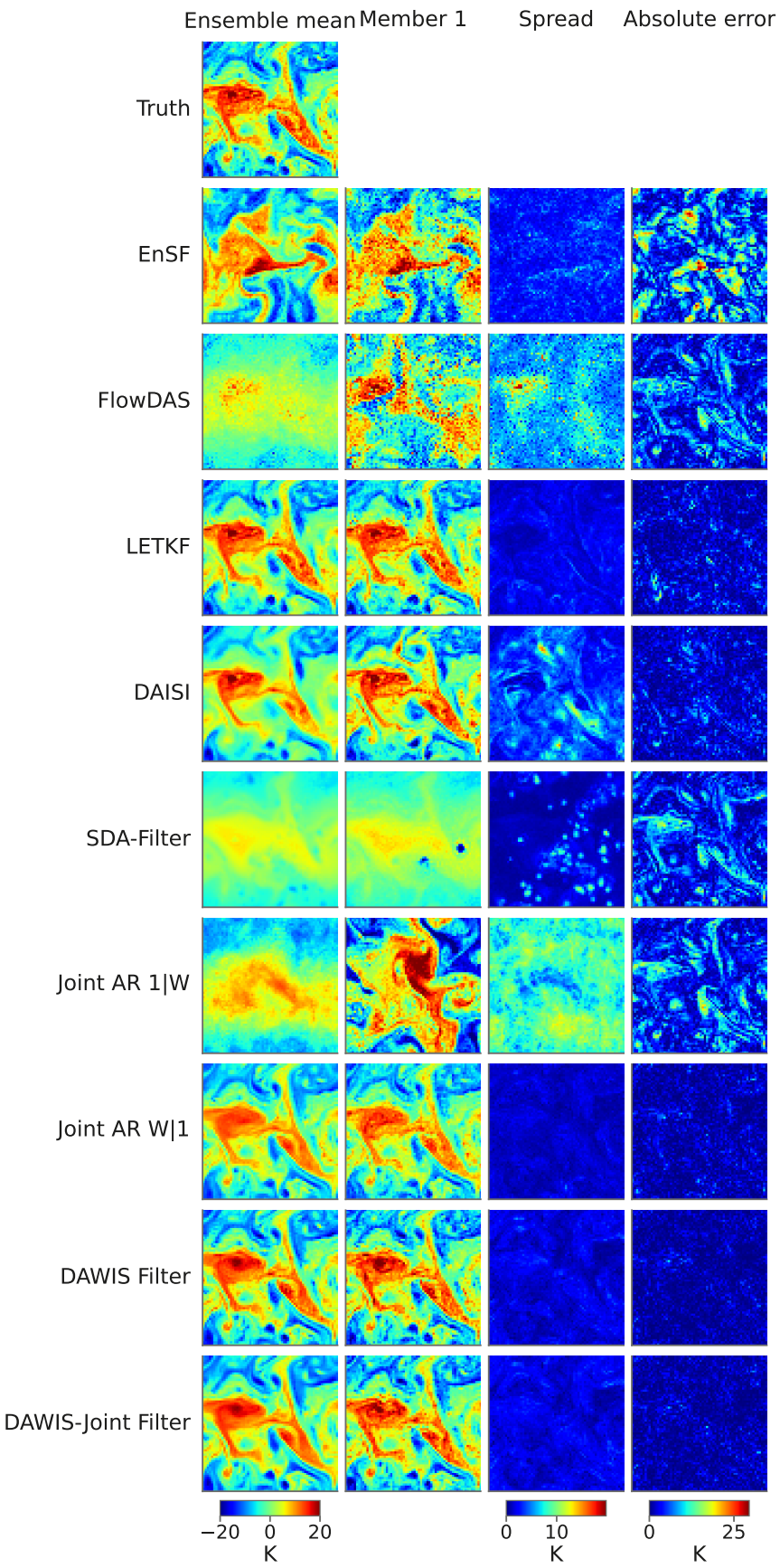


Figure 26: Filtering states at the end of the trajectory for SQG `Saturating`.

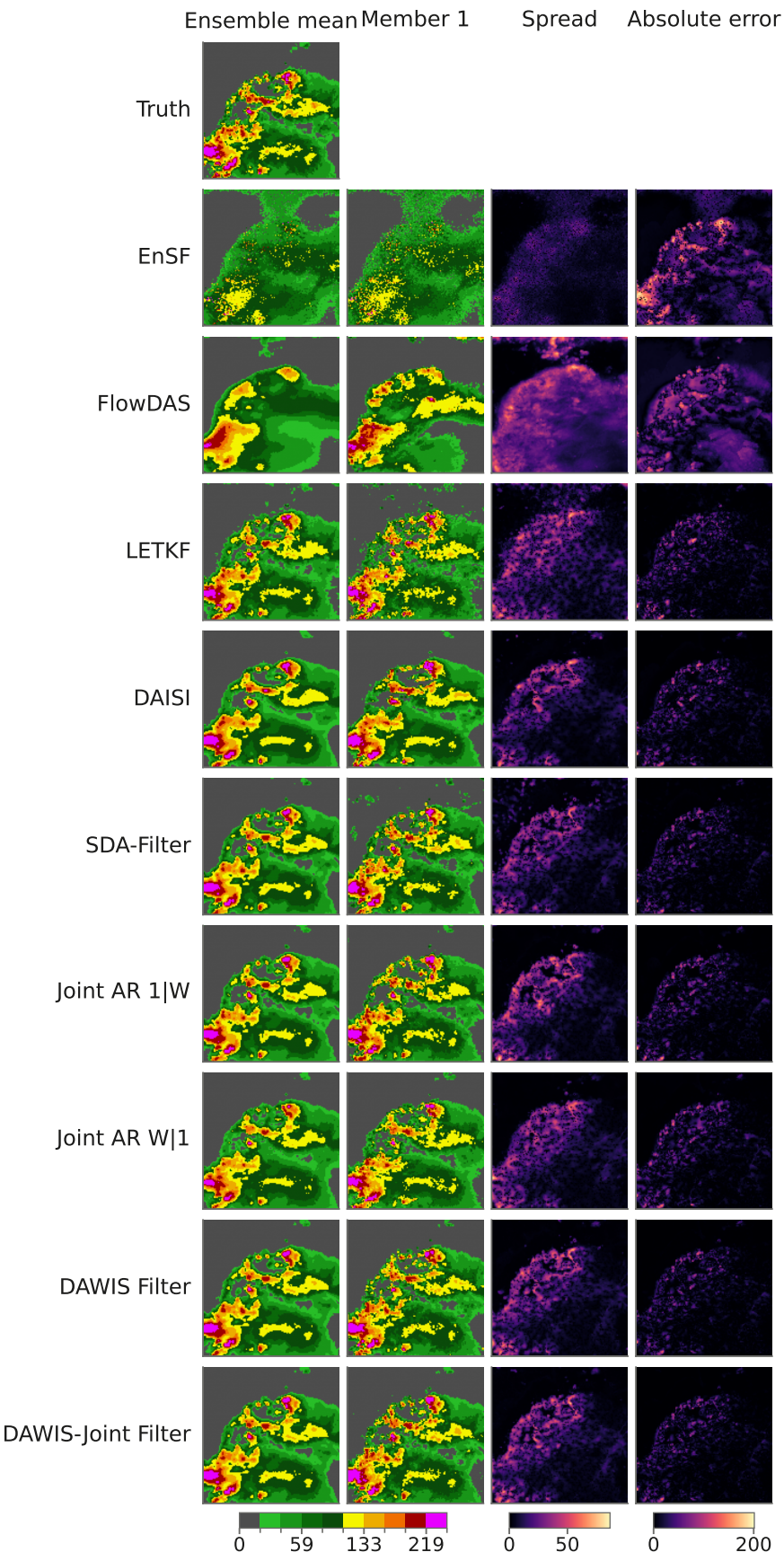


Figure 27: Filtering states at the end of the trajectory for SEVIR.

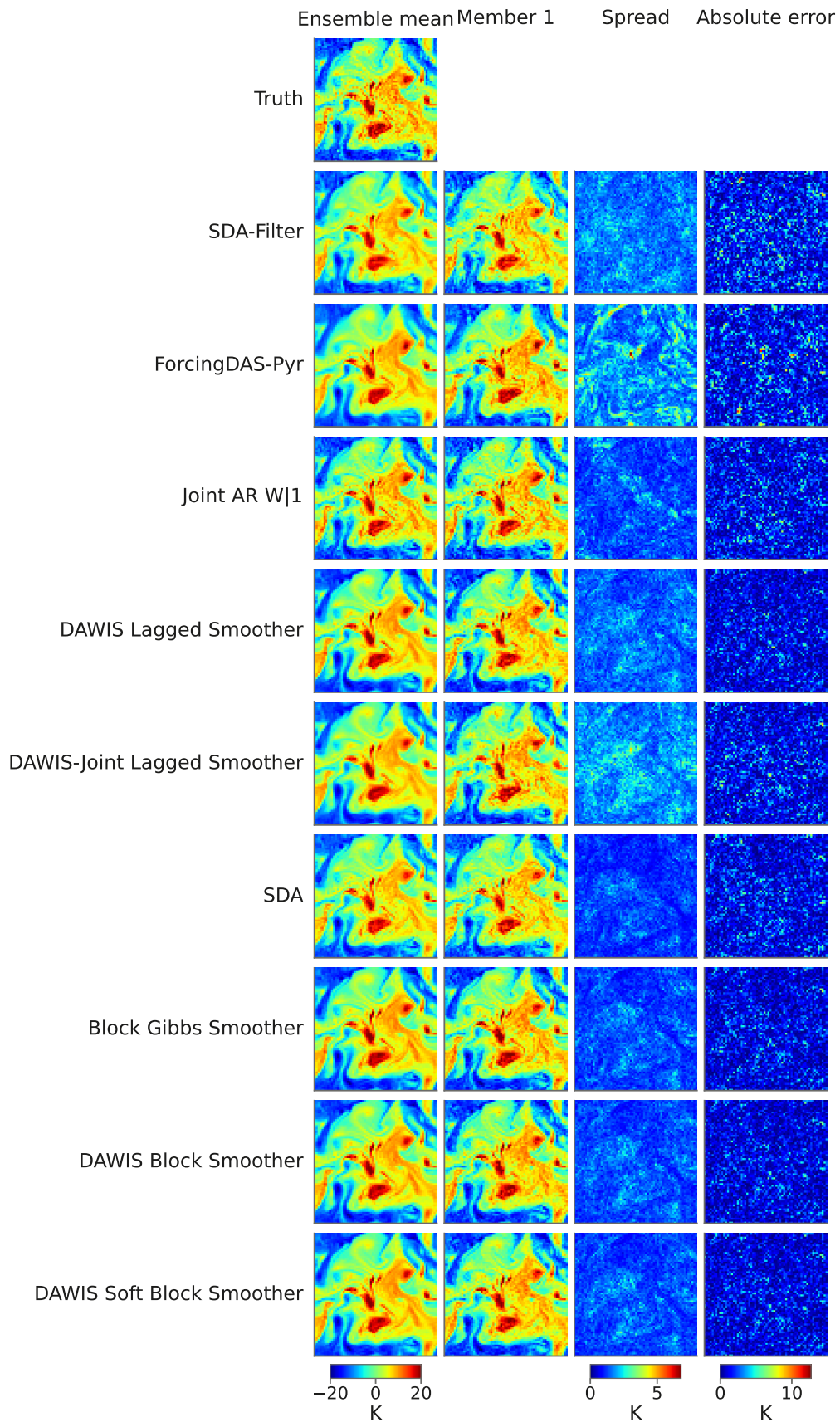


Figure 28: Smoothing states halfway through the trajectory for SQG `Noisy`.

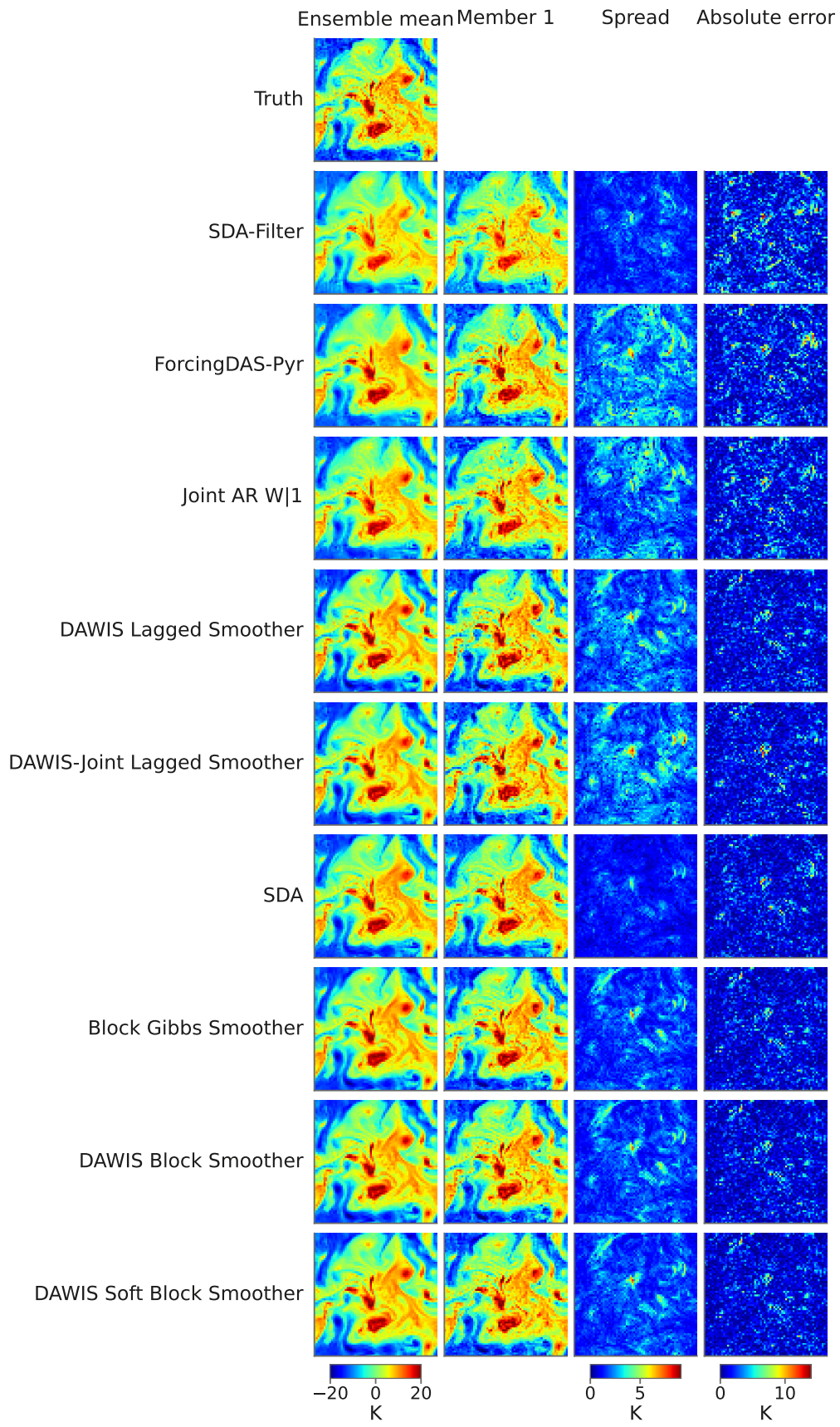


Figure 29: Smoothing states halfway through the trajectory for SQG `Sparse`.

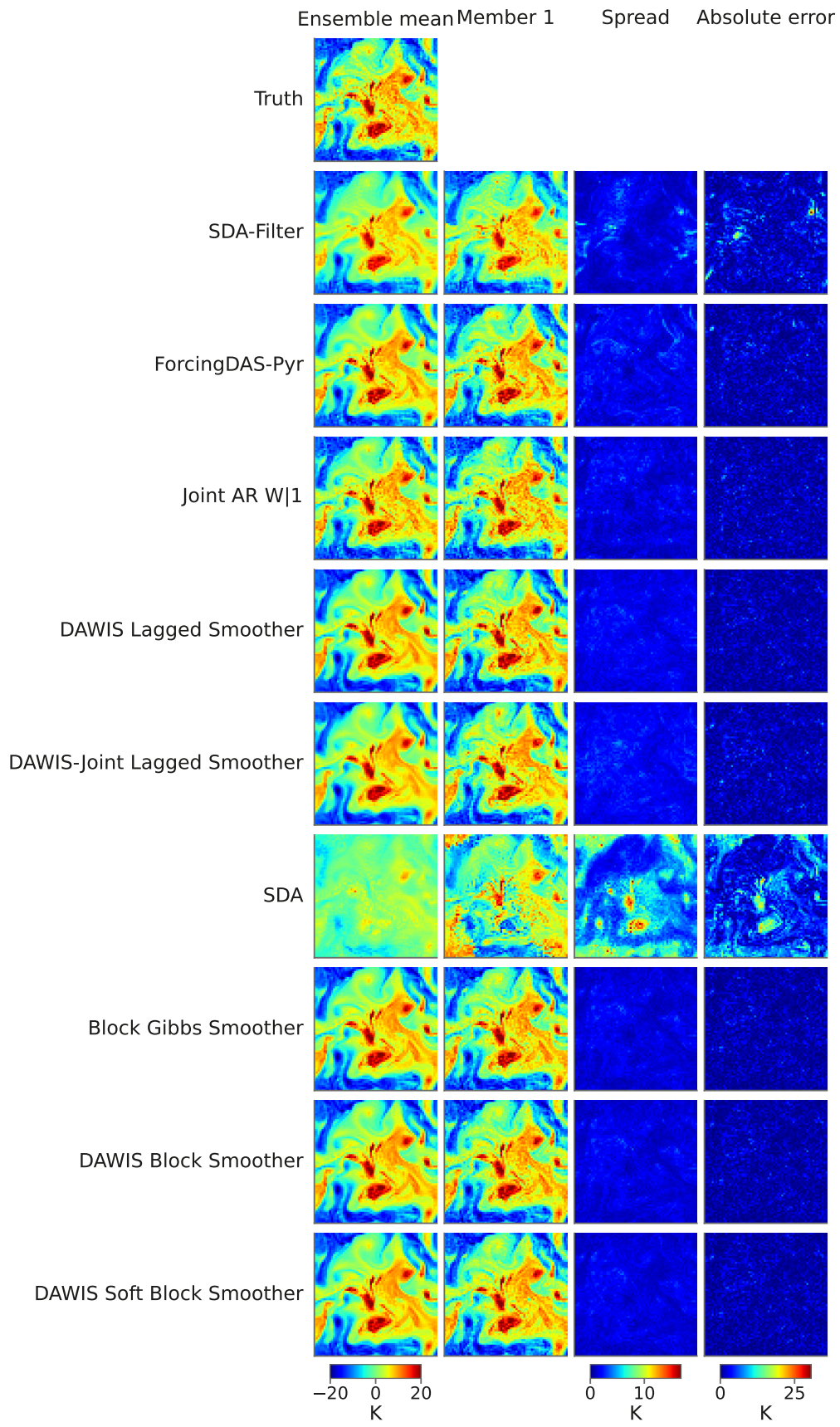


Figure 30: Smoothing states halfway through the trajectory for SQG `Multimodal`.

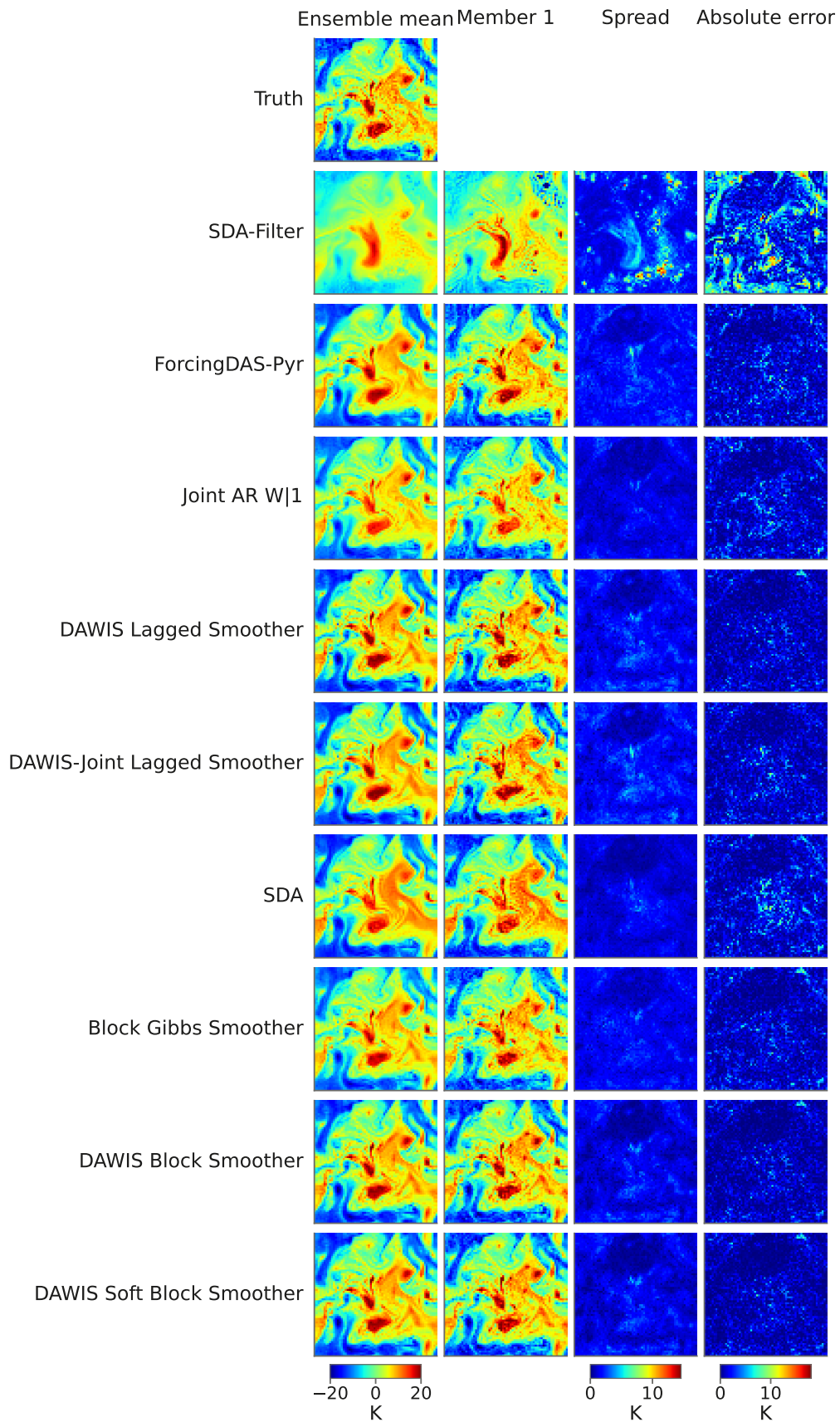


Figure 31: Smoothing states halfway through the trajectory for SQG `Saturating`.

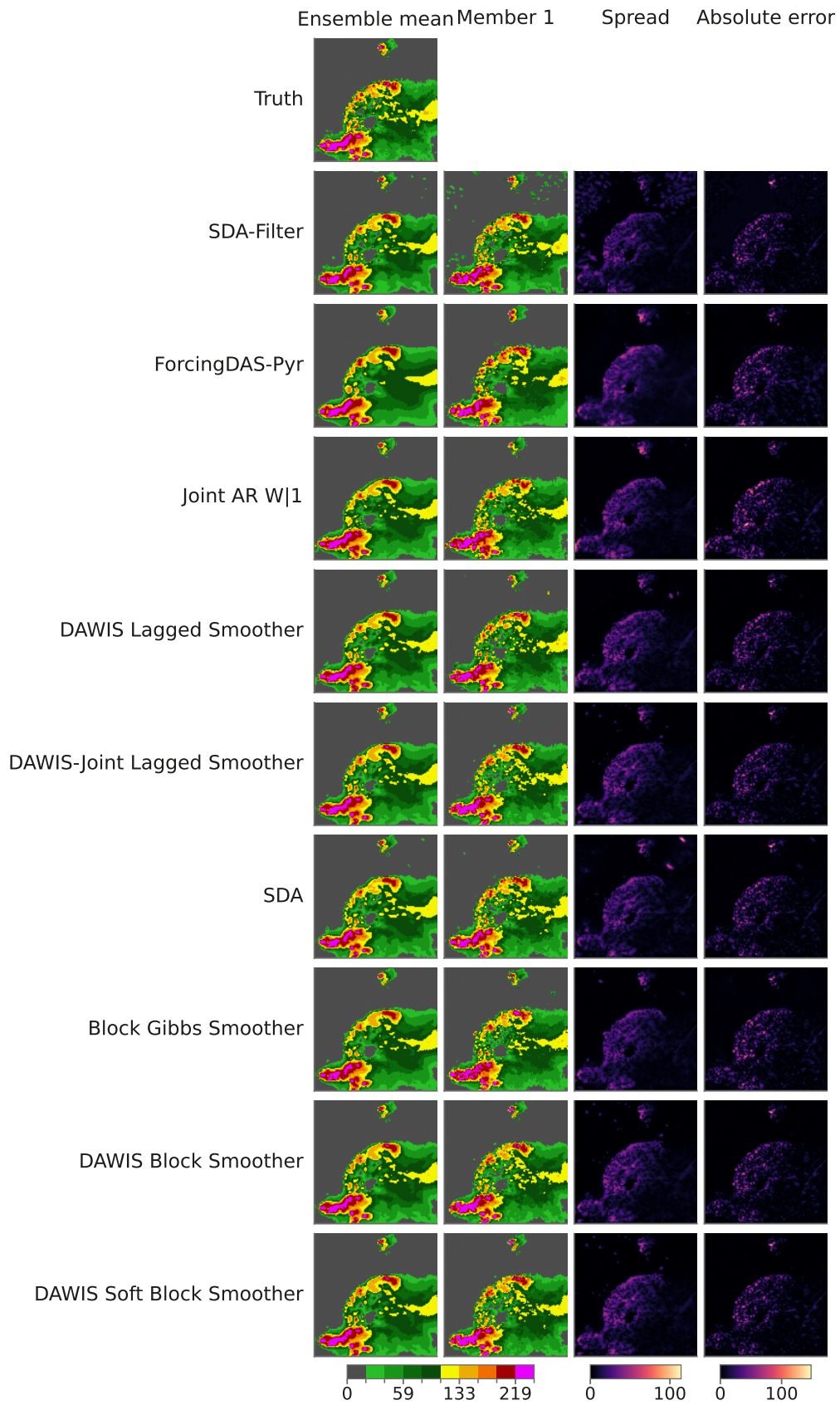


Figure 32: Smoothing states halfway through the trajectory for SEVIR.

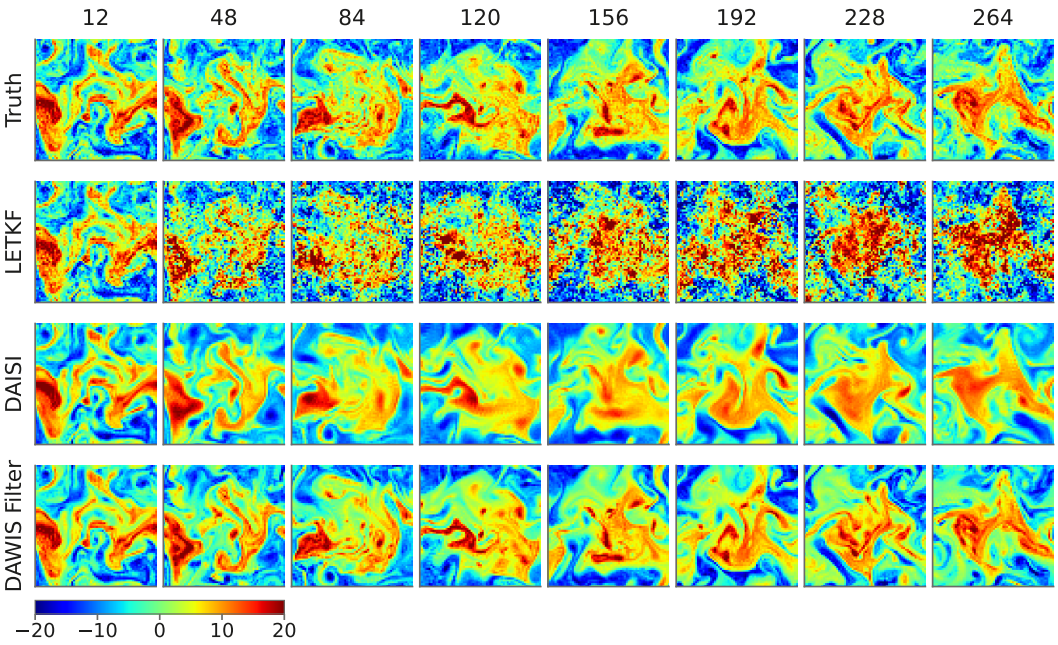


Figure 33: The evolution of the ensemble mean of the filtering distribution over the assimilation window for SQG `Multimodal`.