ObsConDA: Observability-Constrained Data Assimilation with Control-Space Inference

Fang HE[1,3]* **Jiaqi FAN**[1]* **Dan WANG**[2]
[1]Department of Computing, Hong Kong Polytechnic University
[2]The Hong Kong University of Science and Technology
[3]Wuhan University of Technology
fanhe@polyu.edu.hk, jq898.fan@connect.polyu.hk, wangdan@ust.hk

Abstract

Irregular data assimilation is difficult because sparse, off-grid observations do not constrain the same state corrections from one cycle to the next. As the observation geometry changes, the set of supported correction directions changes as well. Many learning-based DA methods infer updates in a fixed latent or control space, even when the current observation layout supports a different set of correction directions. To address this mismatch, we introduce **Obs**ervability-**Con**strained **D**ata **A**ssimilation with Control-Space Inference (ObsConDA), which determines those supported correction directions from the current background and observation geometry and infers only their control coefficients. On real surface-station assimilation, ObsConDA gives the best held-out station accuracy and the best full-field reconstruction among the tested classical and learned baselines. The same advantage carries to matched synthetic observations, explicit geometry shifts, and synthetic dynamical systems, supporting the view that the usable correction directions must change from cycle to cycle. Ablations show that the gain comes from adapting those directions themselves rather than from adding a latent corrector alone.

1 Introduction

Data assimilation (DA) estimates an analysis state from a background forecast and an irregular set of observations [2, 6, 21]. At one assimilation cycle, this is a posterior inference problem [35, 1, 41]:

$$p(\mathbf{x} \mid \mathbf{x}_b, \mathbf{v}, \mathcal{G}) \propto p(\mathbf{v} \mid \mathbf{x}, \mathcal{G})\, p(\mathbf{x} \mid \mathbf{x}_b), \tag{1}$$

where $\mathbf{x}_b$ is the background forecast, $\mathbf{v}$ are observation values, and $\mathcal{G}$ denotes the observation geometry, including locations, variable identities, and reported uncertainties. In geophysical settings, $\mathcal{G}$ is typically sparse, clustered, heterogeneous, and off-grid [28, 27, 37]. The core difficulty is therefore not only that observations are few, but that the geometry of the observation set changes which state corrections are supported by the current observations at that cycle.

This becomes explicit in the local variational form. Writing $\mathbf{x} = \mathbf{x}_b + \delta\mathbf{x}$ and linearizing the observation operator around $\mathbf{x}_b$ gives

$$\delta\mathbf{x}^\star \approx \arg\min_{\delta\mathbf{x}} \frac{1}{2} \left\| J_{\mathcal{G}}\delta\mathbf{x} - \mathbf{r}_{\mathcal{G}} \right\|^2_{R^{-1}} + \frac{1}{2}\|\delta\mathbf{x}\|^2_{B^{-1}}, \tag{2}$$

with residual $\mathbf{r}_{\mathcal{G}} = \mathbf{v} - H_{\mathcal{G}}(\mathbf{x}_b)$. The corresponding local posterior precision [20] is

$$\mathcal{H}_{\mathcal{G}} = B^{-1} + J_{\mathcal{G}}^\top R^{-1} J_{\mathcal{G}}. \tag{3}$$

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

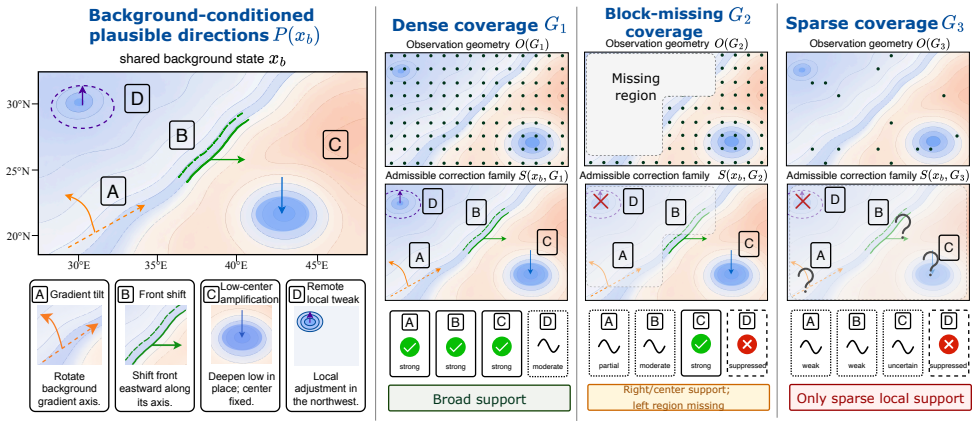

Figure 1: Visual interpretation of the admissible correction family in Eq. (5). Left: a fixed background state induces a shared family $\mathcal{P}(\mathbf{x}_b)$ of locally plausible correction directions. Right: when the background is held fixed and only the observation geometry $\mathcal{G}$ changes, the admissible family contracts to different geometry-dependent subsets. Dense coverage retains a broad set of supported corrections, whereas block-missing and sparse layouts suppress directions that are no longer observation-visible. ObsConDA makes this cycle-wise object explicit by treating $\mathcal{S}(\mathbf{x}_b, \mathcal{G}) \approx \mathcal{P}(\mathbf{x}_b) \cap \mathcal{O}(\mathcal{G})$ as the primary object of inference.

Under irregular coverage, the observation contribution $J_{\mathcal{G}}^{\top} R^{-1} J_{\mathcal{G}}$ in Eq. (3) is often strongly anisotropic and effectively low-rank [12]. Equivalently, after weighting by observation uncertainty, only a small subset of state-space directions is associated with appreciable singular values, so the current observations directly constrain only those directions. A useful update must therefore satisfy two conditions simultaneously: it should remain locally plausible under the background prior, and it should lie in directions that are supported by the current observation geometry.

Equivalently, the local increment decomposes as

$$\delta\mathbf{x} = \delta\mathbf{x}_{\parallel} + \delta\mathbf{x}_{\perp}, \qquad \delta\mathbf{x}_{\parallel} \in \mathrm{span}(V_{\mathcal{G}}^{\mathrm{obs}}), \quad \delta\mathbf{x}_{\perp} \in \ker(J_{\mathcal{G}}), \tag{4}$$

where $\mathrm{span}(V_{\mathcal{G}}^{\mathrm{obs}})$ denotes the observation-visible subspace under the current geometry, while $\ker(J_{\mathcal{G}})$ is the observation-null subspace whose directions do not change the current observations to first order. Accordingly, $\delta\mathbf{x}_{\parallel}$ is the part of the update that the current observations can directly constrain, whereas $\delta\mathbf{x}_{\perp}$ is invisible to them. Irregular-observation DA is therefore fundamentally about restricting inference to a geometry-dependent subset of correction directions, rather than granting equal freedom to all directions in the ambient state space [9].

This reveals the missing object in irregular-observation DA. Before choosing a latent space, a sampler, or a neural posterior map, one must decide which correction directions should even be allowed at the current cycle. We write this cycle-wise object as an admissible correction family

$$\mathcal{S}(\mathbf{x}_b, \mathcal{G}) \approx \underbrace{\mathcal{P}(\mathbf{x}_b)}_{\text{locally plausible directions}} \cap \underbrace{\mathcal{O}(\mathcal{G})}_{\text{geometry-observable directions}}, \tag{5}$$

where $\mathcal{P}(\mathbf{x}_b)$ captures corrections that remain compatible with the local prior geometry around the background state, while $\mathcal{O}(\mathcal{G})$ captures directions that the current observation layout can actually distinguish. In other words, the central question is not merely how to infer an update, but how to parameterize the *support of admissible updates* for the current assimilation cycle. Equation (5) should therefore be read as a cycle-dependent set of supported correction directions, not as a single fixed reduced space used for all assimilation cycles. Figure 1 illustrates this geometry dependence for a fixed background state: as coverage becomes block-missing or sparse, the admissible family contracts to the directions still supported by the current observations.

Existing methods for irregular-observation DA largely fall into three families. Reduced-control and latent variational methods choose a low-dimensional coordinate system and solve DA inside it [3, 40, 11, 42, 39]. Generative methods learn posterior transport from sparse observations to corrected

states [28, 19, 8, 32]. Function-space models improve how irregular geometry is represented [7, 29]. These are all useful advances, but in the settings most relevant here they typically keep the inference space itself fixed across cycles. Latent/control-space methods usually inherit a support before the current observation layout is seen, generative DA usually changes transport over a fixed ambient space, and function-space models usually improve representation without making cycle-wise support restriction itself the object of posterior inference. What remains missing is to treat the cycle-wise admissible correction family in Eqs. (3)–(5) as the primary inference object.

We address this gap with **Obs**ervability-**Con**strained **D**ata **A**ssimilation with Control-Space Inference (ObsConDA). ObsConDA keeps the control-space scaffold of reduced-control and latent variational DA, but changes what is instantiated before the solver runs: rather than solving inside a fixed reduced space or a support inherited from previous cycles, it first builds candidate correction directions from the current background and then restricts them to the directions supported by the current observation geometry. It therefore treats irregular DA as posterior inference over a learned admissible correction family rather than over an unconstrained full state increment. Writing the latent increment as a field over latent coordinate ξ, ObsConDA restricts it to a linear combination of geometry-supported correction directions, $\delta z(\xi) = \sum_{k=1}^{K} \alpha_k \widetilde{\phi}_k(\xi; \mathbf{z}_b, \mathcal{G})$, where $\widetilde{\phi}_k(\xi; \mathbf{z}_b, \mathcal{G})$ are the admissible correction directions retained at the current cycle and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ are their control coefficients. ObsConDA therefore does not optimize an arbitrary latent increment directly; it only infers how strongly to activate those directions. A geometry-conditioned restriction step and control prior define the family passed to the solver, and the coefficients are inferred by a control-space maximum a posteriori (MAP) solver [31]. When the control functions are fixed and the restriction step is the identity, this reduces to classical reduced-space variational DA [15]. Training does not only reconstruct latent analysis increments; it also encourages the learned family to contain the prior-weighted directions that the local inverse problem identifies as observation-relevant.

We validate this support-first formulation across weather DA benchmarks with irregular and shifted observation geometries. In these tests, ObsConDA gives the strongest real-benchmark station and field scores, retains the same advantage under matched synthetic observations and geometry shifts, and loses that advantage when support adaptation is removed.

Contributions. Our reformulation leads to three main contributions. 1) We identify the cycle-wise admissible correction family as the missing object in irregular-observation DA. 2) We introduce ObsConDA, which parameterizes this family with a background-conditioned control-family generator and a geometry-conditioned restriction module before posterior inference is performed. 3) We turn this support-first formulation into a practical control-space variational solver and validate it on real surface-station assimilation, matched synthetic observations, and synthetic dynamical systems, with compact ablations showing that the main gain comes from support adaptation itself.

2 Related Work

Classical and learned control-space DA. Classical DA methods such as 3D-Var, 4D-Var, EnKF, and EnVar [4, 34, 10, 33, 3, 6] define posterior updates through explicit likelihood-prior tradeoffs, but the update family is still determined by fixed covariance structure, hand-designed localization, or globally chosen control variables. Recent learning-based extensions relax these ingredients in different ways. VAE-Var [40], Tensor-Var [42], and LDA [11] learn latent or feature-space coordinates for assimilation, while LoRA-EnVar [39] adapts hybrid ensemble-variational DA with parameter-efficient finetuning. Even when ensemble or hybrid covariances are flow-dependent, these methods still solve inside a support inherited from a latent representation, localized covariance model, or pre-existing control space. A nearby line outside DA proper is likelihood-informed dimension reduction from Bayesian inverse problems [9], which likewise isolates observation-relevant directions, but computes that subspace numerically for each inverse problem instance rather than amortizing it from the current background and observation geometry. ObsConDA stays in this control-space DA line, but re-instantiates that support from the current pair $(\mathbf{x}_b, \mathcal{G})$ before inference begins. The observation geometry therefore does more than condition a regressor or tune a solver state: it decides which correction directions survive and how their coefficients are regularized.

Generative posterior transport for DA. Another major line improves DA by learning richer posterior transport. Score-based DA [26] and DiffDA [19] use diffusion-style priors and observation

guidance for weather-scale assimilation. Latent-EnSF [28] and LD-EnSF [38] combine latent representations with score-based filtering to handle high-dimensional sparse-observation settings, while FlowDAS [8] and Align-DA [32] develop flow-style or aligned posterior transport. Multimodal generative systems such as SLAMS [25] extend this family to heterogeneous station-plus-satellite inputs, but target a different profile-reconstruction setting from the station-driven weather-field assimilation problem studied here. These models differ in how they represent posterior dynamics, but in the present DA setting the learned object is still a score field, transport map, or transition law over a fixed state or latent space. ObsConDA leaves the transport mechanism simple and changes the family over which inference is carried out. In that sense it is complementary to generative DA: the main modeling burden sits in the cycle-wise admissible correction family rather than in a richer posterior transport.

Function-space models under irregular geometry. Semilinear neural operators [29] and FNP [7] show that DA can benefit from function-space models that naturally handle irregular geometry, varying resolution, and continuous spatial structure. More broadly, geometry-aware inverse-operator methods such as GeoFunFlow [36] illustrate the value of defining inference objects over complex geometries rather than committing to a fixed discrete grid. ObsConDA is closest to this line in how it handles off-grid and layout-varying observations, but it does not learn a generic inverse map from observations to states. Instead it uses function-space structure to build background-conditioned candidate correction directions and then restricts them with the current geometry before a control-space MAP solve. This keeps the final inference problem tied to an explicit likelihood-prior objective while letting the supported directions change from cycle to cycle.

3 Method

3.1 Setup

Let $\mathbf{x}_a \in \mathbb{R}^{C\times H\times W}$ denote the target analysis field and let $\mathbf{x}_b \in \mathbb{R}^{C\times H\times W}$ be the background forecast. We observe an irregular collection $\mathcal{Y} = \{(v_i, p_i, \tau_i, c_i)\}_{i=1}^N$, where v_i is the observation value, $p_i \in \mathbb{R}^2$ is the location, $\tau_i > 0$ is the reported uncertainty, and c_i is the observed variable or modality. We split $\mathcal{Y}$ into observation values $\mathbf{v} = (v_1, \dots, v_N)$ and observation geometry $\mathcal{G} = \{(p_i, \tau_i, c_i)\}_{i=1}^N$. We use an encoder-decoder pair (E, D) to define a latent field workspace, with $\mathbf{z}_a = E(\mathbf{x}_a)$ and $\mathbf{z}_b = E(\mathbf{x}_b)$ in $\mathbb{R}^{C_z\times H_z\times W_z}$. We interpret these tensors as samples of latent fields on a latent spatial domain Ω_z with latent grid $\Xi = \{\xi_m\}_{m=1}^M$.

3.2 Method Overview

ObsConDA separates three questions that are usually entangled in latent DA: 1) which correction directions are plausible around the current background, 2) which of those directions are supported by the current observation geometry, and 3) how strongly the surviving directions should be activated. Figure 2 summarizes this decomposition. The same trained ObsConDA model is reused across cycles: when the observation layout changes, we re-instantiate the admissible family and control prior from the current pair $(\mathbf{x}_b, \mathcal{G})$ rather than retraining a separate model for each geometry regime.

Cycle-wise admissible correction family. At each cycle, ObsConDA first decides which latent directions are even worth searching. Because observation geometry changes from cycle to cycle, the set of supported corrections changes as well. Around the current background state, the local latent inverse problem can be written as

$$\delta\mathbf{z}^\star \approx \arg\min_{\delta\mathbf{z}} \frac{1}{2}\,\|\mathcal{J}_\mathcal{G}\delta\mathbf{z} - \mathbf{r}_\mathcal{G}\|^2_{R^{-1}} + \frac{1}{2}\|\delta\mathbf{z}\|^2_{\mathcal{B}^{-1}}, \tag{6}$$

where $\mathbf{r}_\mathcal{G} = \mathbf{v} - H_\mathcal{G}(D(\mathbf{z}_b))$, $\mathcal{J}_\mathcal{G}$ is the local latent observation Jacobian, and $\mathcal{B}$ is the local latent prior covariance. Under sparse and heterogeneous coverage, the observation term is typically low-rank and anisotropic, so only a geometry-dependent subset of latent directions is directly constrained. We denote the corresponding latent target by the cycle-wise admissible family $\mathcal{S}(\mathbf{z}_b, \mathcal{G})$. A useful local proxy is the span of prior-weighted observation backprojections, namely the column space of $\mathcal{B}\mathcal{J}_\mathcal{G}^\top$: $\mathcal{J}_\mathcal{G}^\top$ brings observation-space signals back into latent directions, and $\mathcal{B}$ reshapes them according to the local prior geometry. ObsConDA approximates this object constructively in two stages. The background branch builds a candidate family together with a low-rank plausibility surrogate $\widehat{\mathcal{B}}(\mathbf{z}_b)$,

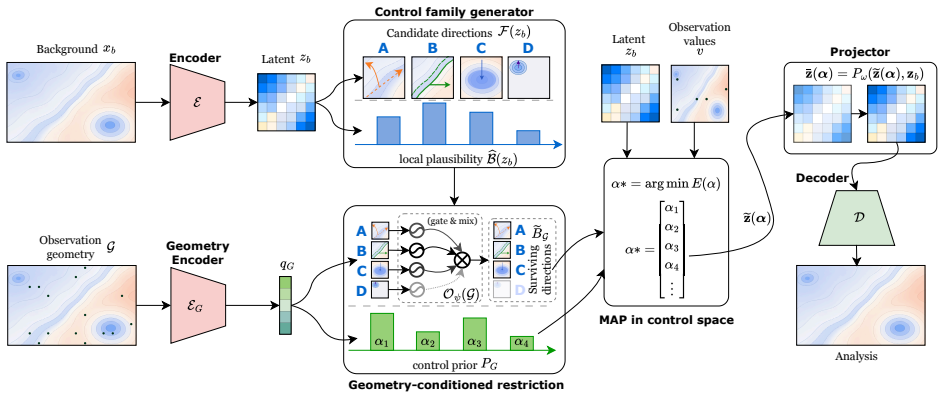

Figure 2: Overview of ObsConDA. The *Control-family generator* uses the latent background $\mathbf{z}_b$ to produce candidate correction directions and their local plausibility structure. The *Geometry-conditioned restriction* module uses the current geometry to keep only the directions supported at the present cycle and to instantiate the corresponding control prior over their coefficients. *MAP in control space* then infers only the coefficient vector $\boldsymbol{\alpha}^\star$ over this restricted family, after which the resulting latent proposal is projected by P_ω and decoded by D to obtain the final analysis.

and the geometry branch restricts that candidate family into an admissible correction family $\widetilde{\mathcal{F}}(\mathbf{z}_b, \mathcal{G})$ and its associated control prior. Inference therefore takes place inside a cycle-specific latent family rather than in the ambient latent space. Training also includes a support-alignment loss that asks this restricted family to do two things at once: reconstruct realistic latent analysis increments and contain the prior-weighted observation backprojections induced by the current geometry. This is the training signal that aligns the learned family with observation-visible directions before the final solver is unrolled. Proposition 3 in Appendix A.1 formalizes the role of prior-weighted observation backprojections as the local support target.

Control-family generator. The background state and the observation geometry play different roles in ObsConDA, so the final family is built in two stages. The *Control-family generator* maps the current latent background to a candidate family $\mathcal{F}(\mathbf{z}_b) = \mathrm{span}\{\phi_1(\cdot; \mathbf{z}_b), \ldots, \phi_K(\cdot; \mathbf{z}_b)\}$ of locally plausible correction directions, together with a nonnegative spectrum that induces the local prior surrogate $\widehat{\mathcal{B}}(\mathbf{z}_b)$. This surrogate does not yet encode the current observation layout. It only summarizes which directions remain background-locally plausible and supplies the prior geometry used during training to form prior-weighted observation backprojections.

Geometry-conditioned restriction. The *Geometry-conditioned restriction* module then maps the observation geometry $\mathcal{G}$ to a context vector $q_\mathcal{G}$ and uses lightweight heads to produce both an admissibility operator $\mathcal{O}_\psi(\mathcal{G}) : \mathcal{F}(\mathbf{z}_b) \to \widetilde{\mathcal{F}}(\mathbf{z}_b, \mathcal{G})$ and a geometry-aware control prior $P_\mathcal{G}$. The operator acts on the candidate family and yields the cycle-wise admissible correction family actually passed to the solver. The control prior plays a different role: once inference has been restricted to the surviving directions, it regularizes only how strongly their coefficients may be activated. In other words, the restriction step decides which directions survive at the current cycle, while the geometry-aware prior acts only after this restriction and shapes the coefficient distribution on those surviving directions. Writing $\widetilde{\phi}_k(\cdot; \mathbf{z}_b, \mathcal{G}) = \mathcal{O}_\psi(\mathcal{G})[\phi_k(\cdot; \mathbf{z}_b)]$, ObsConDA restricts the latent increment to that admissible family via $\delta z(\xi) = \sum_{k=1}^{K} \alpha_k \widetilde{\phi}_k(\xi; \mathbf{z}_b, \mathcal{G})$ with reduced control variable $\boldsymbol{\alpha} \in \mathbb{R}^K$. Appendix A.1 gives the sampled control matrix, admissibility operator, and geometry-aware control prior used in this finite-rank realization, and records the limiting case in which ObsConDA reduces to classical reduced-control DA when the control functions are fixed and the restriction and projection steps collapse to the identity.

MAP in control space. Once the admissible correction family has been fixed, inference reduces to choosing only the activation strengths of its surviving directions. ObsConDA therefore solves a low-dimensional MAP problem over control coefficients rather than over an unconstrained latent increment. The solver receives the pair $(\widetilde{B}_\mathcal{G}, P_\mathcal{G})$. Every admissible latent proposal then

takes the form $\widetilde{\mathbf{z}}(\boldsymbol{\alpha}) = \mathbf{z}_b + \widetilde{B}_{\mathcal{G}}\boldsymbol{\alpha}$. To keep the optimized state on the learned latent manifold, ObsConDA applies a background-conditioned latent projector inside the objective and optimizes over $\bar{\mathbf{z}}(\boldsymbol{\alpha}) = P_\omega(\widetilde{\mathbf{z}}(\boldsymbol{\alpha}), \mathbf{z}_b)$. The corresponding projector-based physical-structure regularizer is $\mathcal{R}_{\text{phys}}(\widetilde{\mathbf{z}}, \mathbf{x}_b) = \|\widetilde{\mathbf{z}} - P_\omega(\widetilde{\mathbf{z}}, \mathbf{z}_b)\|_2^2$. It is not an explicit PDE or conservation-law constraint; instead, it keeps the proposal close to a learned latent workspace whose training preserves multiscale spectra, cross-variable consistency, and short-horizon forecast structure. ObsConDA then infers the control coefficients by solving the control-space MAP problem

$$\mathcal{E}(\boldsymbol{\alpha}) = \frac{1}{2}\left\|H_{\mathcal{G}}(D(\bar{\mathbf{z}}(\boldsymbol{\alpha}))) - \mathbf{v}\right\|_{R^{-1}}^2 + \frac{1}{2}\boldsymbol{\alpha}^\top P_{\mathcal{G}}\boldsymbol{\alpha} + \lambda_{\text{phys}}\mathcal{R}_{\text{phys}}\big(\widetilde{\mathbf{z}}(\boldsymbol{\alpha}), \mathbf{x}_b\big), \tag{7}$$

where $P_{\mathcal{G}}$ is the geometry-aware control prior generated above, represented here as a quadratic precision term over $\boldsymbol{\alpha}$, and R is the observation error covariance. The three terms correspond to observation fit, geometry-aware control regularization, and projector-based regularization toward fields that remain consistent with the learned physical structure. The same projected latent state is used both inside the observation term and at output time, so projection is part of the optimized forward map rather than a post-hoc cleanup step. The next subsection specifies the resulting unrolled control-space solver. Training uses complementary surrogates for latent-increment plausibility and prior-weighted innovation alignment; Appendix A.3 gives the staged ObsConDA training recipe together with the exact support-alignment equations.

3.3 Unrolled Control-Space Solver

Eq. (7) is optimized with unrolled damped Gauss-Newton steps [24] in the K-dimensional control space rather than in the ambient latent state. At iterate $\boldsymbol{\alpha}_t$, we first form the current admissible proposal $\widetilde{\mathbf{z}}_t = \widetilde{\mathbf{z}}(\boldsymbol{\alpha}_t)$, project it to $\bar{\mathbf{z}}_t = P_\omega(\widetilde{\mathbf{z}}_t, \mathbf{z}_b)$, and compute the residual $\mathbf{r}_t = \mathbf{v} - H_{\mathcal{G}}(D(\bar{\mathbf{z}}_t))$. We then linearize the projected decoded observation map w.r.t. the control coordinates and solve

$$\left(J_t^\top R^{-1} J_t + P_{\mathcal{G}} + \lambda_t I\right)\Delta\boldsymbol{\alpha}_t = J_t^\top R^{-1}\mathbf{r}_t - P_{\mathcal{G}}\boldsymbol{\alpha}_t - \lambda_{\text{phys}}\nabla_{\boldsymbol{\alpha}}\mathcal{R}_{\text{phys}}(\widetilde{\mathbf{z}}_t, \mathbf{x}_b), \tag{8}$$

where $J_t = \nabla_{\boldsymbol{\alpha}} H_{\mathcal{G}}(D(\bar{\mathbf{z}}(\boldsymbol{\alpha}_t)))$ is the control-space Jacobian of the projected decoded observation map. The iterate is then updated as $\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \beta_t\Delta\boldsymbol{\alpha}_t$. The residual term asks which admissible directions improve data fit after projection to the learned manifold, the prior term suppresses overly aggressive control activations, and the physical-structure regularizer penalizes proposals that require large projector corrections. Around the converged iterate, the same control-space structure also admits a Laplace approximation for local posterior uncertainty; Appendix A.2 gives the corresponding covariance, sampling formula, and complexity discussion. Appendix A.4 gives the step-by-step pseudocode for the final test-time inference pipeline.

3.4 Interpretation of the Support-First Formulation

Cycle-dependent supported directions. Sparse, clustered, and heterogeneous observations constrain only part of the latent space at any given cycle. Instantiating the admissible correction family from the current background and observation geometry makes this restriction explicit: the solver is allowed to search only directions that are locally plausible and supported by the current observation layout, rather than treating the entire latent space as uniformly available.

Separated roles of background and geometry. ObsConDA does not ask one module to absorb background state, observation geometry, and posterior fitting all at once. The background branch models which directions remain locally plausible, while the geometry branch determines which of those directions are supported at the current cycle and how strongly their coefficients should be regularized. This separation keeps the source of each inductive bias explicit: background state governs plausibility, and observation geometry governs support and regularization.

Low-dimensional posterior inference. Once $\widetilde{B}_{\mathcal{G}}$ and $P_{\mathcal{G}}$ are fixed for the current cycle, the remaining inference problem is only over the control coefficients. This keeps posterior inference low-dimensional and second-order, while preserving a clear connection to classical reduced-control variational DA as the special case where the admissibility step collapses to the identity.

Table 1: Track A real surface-station assimilation results.

Method	Held-out NRMSE ↓	Held-out NCRPS ↓	90% Cov. Err. ↓	Full-field NRMSE ↓	Rel. Improve ↑
Background	0.1241	0.0698	0.0613	0.0327	0.000
IDW/OI	0.1164	0.0621	0.0630	0.0326	0.003
3DVar	0.1152	0.0632	**0.0479**	0.0325	0.006
FNP	0.1171	0.0663	0.0618	0.0326	0.003
VAE-Var	0.1170	0.0662	0.0614	0.0324	0.008
ObsConDA	**0.1091**	**0.0611**	0.0493	**0.0272**	**0.167**

4 Experiments and Results

We evaluate ObsConDA on two main data-assimilation benchmark tracks. To assess practical data-assimilation performance under real observational irregularity, Track A uses raw surface-station reports with their native coverage, density, and missingness patterns. To isolate algorithmic behavior under matched observation noise and controlled geometry shifts, Track B uses the same underlying weather states but replaces the raw station reports with synthetic observations sampled from the target analysis fields. To measure the contribution of the main ObsConDA components, we also report compact ablations. To check whether the same support-first behavior persists on synthetic dynamical systems, Appendix C reports additional validation on Lorenz-96, a 2D PDE benchmark, and a compact Lorenz-63 sanity check.

4.1 Track A: Real Surface-Station Assimilation

Dataset and setup. Each Track A cycle pairs an Aurora global Earth-system forecast [5] with an ERA5 single-level analysis target [17] and raw surface reports from the NOAA Integrated Surface Database (ISD) [30]. We focus on the four primary near-surface variables $u10$, $v10$, $t2m$, and msl. The data span 2024-08-01 to 2024-12-31, with August–October used for training, November for validation, and December for test; all cycles are evaluated at 00/06/12/18 UTC with a 30-minute observation window.

Native ISD irregularity is kept throughout, including nonuniform station placement, variable-dependent availability, and cycle-to-cycle missingness. Each cycle assimilates up to 3072 stations and reserves another 1024 for held-out evaluation, stratified over geography and variable availability. We compare ObsConDA against the background forecast, IDW/OI, 3DVar, FNP [7], and VAE-Var [40]. IDW/OI and 3DVar assimilate the irregular station values directly. FNP and VAE-Var instead first align the observation information with their field representation through their own interpolation or observation-alignment interface before producing the comparison fields on the same cycles and held-out splits. The benchmark cycles, train/validation/test split, and held-out evaluation protocol are therefore fixed across methods, while each baseline keeps its native observation interface.

Metrics. We evaluate Track A with normalized RMSE, normalized CRPS, coverage error, and relative improvement over the background. Precise metric definitions appear in Appendix B.1.

Results. ObsConDA gives the lowest held-out NRMSE and NCRPS on the real benchmark, reducing them to 0.1091 and 0.0611 from 0.1152 and 0.0632 for 3DVar and 0.1170 and 0.0662 for VAE-Var. The larger gain appears in the full field, where it reaches 0.0272 NRMSE versus 0.0327 for the background and 0.0324 for VAE-Var. The only row it does not lead is 90% coverage error, where 3DVar is slightly better at 0.0479 versus 0.0493. Together these results show that ObsConDA turns irregular real observations into better held-out station accuracy and markedly stronger full-field reconstruction. We leave the per-variable breakdowns, a representative qualitative case, and the station-budget scaling check to Appendix B.2.

4.2 Track B: Simulated Observations on Real Weather Fields

Dataset and setup. Track B uses the same background–target pairs, variables, and chronological split as Track A, but replaces the raw ISD reports with synthetic observations sampled from the target analysis fields and perturbed by known variable-wise noise. We compare the background forecast, 3DVar, FNP, VAE-Var, FlowDAS [8], and ObsConDA, with FlowDAS as the transport-style

Table 2: Track B matched-observation results on real weather fields.

Method	Surface NRMSE ↓	Surface NCRPS ↓	90% Cov. Err. ↓	Geometry-OOD NRMSE ↓	Geometry-OOD NCRPS ↓
Background	0.0318	0.0286	0.0963	0.0312	0.0284
3DVar	0.0305	0.0278	0.0975	0.0297	0.0274
FNP	0.0303	0.0276	0.0962	0.0295	0.0273
VAE-Var	0.0301	0.0277	0.0964	0.0293	0.0272
FlowDAS	0.0308	0.0281	**0.0794**	0.0307	0.0280
ObsConDA	**0.0258**	**0.0268**	0.0966	**0.0252**	**0.0266**

Table 3: Track B geometry-shift and sparsity stress test.

Method	Seen NRMSE ↓	Block-missing NRMSE ↓	Mixed-density NRMSE ↓	Random→Real Gap ↓	10% Obs NRMSE ↓	30% Obs NRMSE ↓	50% Obs NRMSE ↓
3DVar	0.0329	0.0331	0.0268	0.0037	0.0510	0.0496	0.0482
FNP	0.0314	0.0316	0.0256	0.0032	0.0496	0.0481	0.0469
VAE-Var	0.0311	0.0313	0.0253	0.0031	0.0492	0.0478	0.0466
FlowDAS	0.0318	0.0320	0.0259	0.0034	0.0500	0.0487	0.0475
ObsConDA w/ fixed support	0.0446	0.0446	0.0356	0.0053	0.0622	0.0607	0.0591
ObsConDA w/o support loss	0.0451	0.0453	0.0362	0.0050	0.0632	0.0616	0.0600
ObsConDA	**0.0276**	**0.0275**	**0.0231**	**0.0028**	**0.0472**	**0.0459**	**0.0450**

generative baseline. As in Track A, 3DVar operates directly on the observation layout, while FNP, VAE-Var, and FlowDAS first align the observation information with their field representation through their own interpolation or observation-alignment interface before producing the comparison fields on the same synthetic cycles. The matched benchmark and evaluation rows are therefore fixed across methods, while each baseline keeps its native observation interface.

We evaluate four observation layouts. `real_station` uses the same station coordinates from the corresponding real cycle. `random_irregular` replaces that layout with a count-matched irregular point cloud sampled over the same domain. `block_missing` removes one contiguous rectangular region from the real-station layout. `mixed_density` combines a dense local cluster with a sparser background pattern. The same trained ObsConDA model is used across all geometry families.

Metrics. We report surface NRMSE, surface NCRPS, 90% coverage error, geometry-OOD NRMSE, and geometry-OOD NCRPS. Surface metrics average over all four geometry families. Geometry-OOD metrics average only over the three shifted families `random_irregular`, `block_missing`, and `mixed_density`. Table 3 reports the corresponding seen-geometry, geometry-shift, and sparse-observation rows.

Results. ObsConDA is best on all four main rows. Over the full geometry mix, it reduces surface NRMSE to 0.0258 from 0.0301 for VAE-Var and 0.0303 for FNP, and lowers surface NCRPS to 0.0268 from 0.0276 and 0.0277. The same pattern holds on the shifted families alone, where geometry-OOD NRMSE drops to 0.0252 from 0.0293 for VAE-Var and 0.0295 for FNP, while geometry-OOD NCRPS falls to 0.0266 from 0.0272 and 0.0273. This is a strong-prior benchmark, with several learned baselines staying close to the no-assimilation reference, but ObsConDA still turns the synthetic observations into the clearest net gain. FlowDAS remains best on the 90% coverage row at 0.0794, but that calibration advantage does not carry to the NRMSE and NCRPS rows. Representative qualitative Track B cases are collected in Appendix B.3.

Geometry shift and support learning. ObsConDA is best in every column, not only on the seen layout but also under block missingness, mixed density, random-to-real transfer, and all three sparse-observation levels. The gap to FNP, VAE-Var, and FlowDAS is steady, but the gap to the ablations is much larger. Once the admissible correction family stops adapting to the current geometry, seen NRMSE jumps from 0.0276 to about 0.045, block-missing and mixed-density errors rise into the 0.036–0.045 range, and the same degradation carries through the 10%, 30%, and 50% rows. Once the allowed correction family no longer tracks the observation layout, the gains collapse.

Additional Track B cases under matched and shifted geometries are collected in Appendix B.3. We defer the corresponding comparison of probabilistic scores and interval calibration to Appendix B.4. Runtime and GPU-memory measurements are summarized in Appendix B.5.

4.3 ObsConDA Ablations

Setup. To isolate which parts of ObsConDA carry the support-first gain, Table 4 removes one component at a time. Across all variants we keep the same data splits, latent workspace, solver depth,

Table 4: ObsConDA ablations on Track A and Track B.

Method	Track A Held-out NRMSE ↓	Track A Held-out NCRPS ↓	Track B Geometry-OOD NRMSE ↓	Track B Norm. Cal. Err. ↓
ObsConDA	**0.109**	**0.061**	**0.032**	**0.049**
ObsConDA w/ fixed support	0.127	0.072	0.063	0.102
ObsConDA w/o support loss	0.127	0.072	0.065	0.102
ObsConDA w/o geometry-conditioned prior	0.127	0.072	0.067	0.102
ObsConDA w/ post-hoc projection	0.136	0.072	0.081	0.092
ObsConDA w/o projector	0.129	0.072	0.059	0.102

and evaluation protocol, and change only the named support or projection component. 'ObsConDA w/ fixed support' freezes the admissible correction family across cycles. 'ObsConDA w/o support loss' removes the training signal that aligns that family with observation-visible directions. 'ObsConDA w/o geometry-conditioned prior' keeps the solver but drops the geometry dependence from the prior on the coefficients. 'ObsConDA w/ post-hoc projection' applies the projector only after solving, and 'ObsConDA w/o projector' removes it entirely.

Results. Freezing the support is the first clear failure mode: Track A moves from 0.109/0.061 to 0.127/0.072, and Track B geometry-OOD NRMSE nearly doubles from 0.032 to 0.063. Removing the support loss produces almost the same drop, with Track B rising to 0.065, which shows that the support parameterization and the training signal that aligns it with observation-visible directions matter together. Dropping the geometry-conditioned prior is slightly worse again at 0.067 on Track B, so the prior helps, but the larger loss has already happened once support stops following the current geometry. Projection timing matters in a different way. Post-hoc projection gives the worst errors, reaching 0.136 and 0.081, whereas removing the projector entirely is less damaging on geometry-OOD error at 0.059. The strongest version is therefore not just a latent corrector with a projection step; it is the combination of cycle-wise support adaptation and in-loop projection.

5 Limitations

ObsConDA is scoped to a specific but practically relevant DA regime. The present study focuses on archived forecast–analysis workflows with irregular surface observations, short assimilation windows, and geometry shifts of the kind studied in Tracks A and B. The main boundaries are the source of supervision, the observation model, and the range of geometry shifts covered here. **1) Forecast–analysis supervision.** The current training recipe assumes access to forecast–analysis pairs, or to an equivalent DA archive from which latent analysis increments can be formed. This matches reanalysis-style and archived operational settings, which are the focus of the present study. **2) Observation-model scope.** Our experiments use station-driven weather DA with short windows and a heteroscedastic Gaussian likelihood with diagonal observation error covariance. This is a reasonable starting point for irregular surface assimilation, but denser multimodal observing systems may require richer observation operators and error models. **3) Generalization beyond the studied geometry shifts.** The same trained ObsConDA model handles the observation layouts studied here, including real-station irregularity, random irregular layouts, block missingness, mixed density, and sparse-observation stress tests. Broader shifts in sensor type, variable mix, spatial footprint, or assimilation horizon are outside the present benchmark and are left for future evaluation.

6 Conclusion

This paper introduced ObsConDA, an observability-constrained control-space inference framework for irregular data assimilation. Rather than leaving the cycle-wise update support implicit, ObsConDA rebuilds an admissible correction family from the current background and observation geometry, and then performs posterior inference only over the corresponding control coefficients. Across real surface-station assimilation, matched synthetic observation benchmarks, geometry-shift stress tests, and synthetic dynamical systems, this support-first construction consistently improves held-out accuracy, full-field reconstruction, and robustness under changing observation layouts. The ablations show that the gain is tied to support adaptation itself: once the correction family no longer tracks the current geometry, the advantage largely disappears. These results suggest that making admissible update support explicit is a useful organizing principle for learned variational assimilation under irregular observations.

Acknowledgments

Dan Wang's work is supported in part by RGC GRF 15230624, 15239925, 16213526, CRF C5020-25GF, ITC ITS/052/23MX, and a start-up fund of HKUST.

References

[1] Asad Aali, Giannis Daras, Brett Levac, Sidharth Kumar, Alex Dimakis, and Jon Tamir. Ambient diffusion posterior sampling: Solving inverse problems with diffusion models trained on corrupted data. In *The Thirteenth International Conference on Learning Representations*, 2025.

[2] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.

[3] 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.

[4] Dale M Barker, Wei Huang, Yong-Run Guo, AJ Bourgeois, and QN Xiao. A three-dimensional variational data assimilation system for mm5: Implementation and initial results. *Monthly Weather Review*, 132(4):897–914, 2004.

[5] Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, et al. A foundation model for the earth system. *Nature*, 641:1180–1187, 2025.

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

[7] Kun Chen, Peng Ye, Hao Chen, Tao Han, Wanli Ouyang, Tao Chen, LEI BAI, et al. Fnp: Fourier neural processes for arbitrary-resolution data assimilation. *Advances in Neural Information Processing Systems*, 37:137847–137872, 2024.

[8] 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.

[9] Tiangang Cui, James Martin, Youssef M. Marzouk, Antti Solonen, and Alessio Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30 (11):114015, 2014.

[10] Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer, 2009.

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

[12] H. P. Flath, Lucas C. Wilcox, Volkan Akçelik, Judith Hill, Bart van Bloemen Waanders, and Omar Ghattas. Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial hessian approximations. *SIAM Journal on Scientific Computing*, 33(1):407–432, 2011.

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

[14] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69 (2):243–268, 2007.

[15] Serge Gratton, Patrick Laloyaux, Annick Sartenaer, and Jean Tshimanga Ilunga. A reduced and limited memory preconditioned approach for the 4d-var data assimilation problem. *Quarterly Journal of the Royal Meteorological Society*, 137:452–466, 2011.

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

[17] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020.

[18] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[19] Langwen Huang, Lukas Gianinazzi, Yuejiang Yu, Peter D Dueben, and Torsten Hoefler. Diffda: a diffusion model for weather-scale data assimilation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 19798–19815, 2024.

[20] Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*. Springer, New York, NY, 2004.

[21] Kody Law, Andrew Stuart, and Kostas Zygalakis. Data assimilation. *Cham, Switzerland: Springer*, 214(52):7, 2015.

[22] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2): 130–141, 1963.

[23] Edward N. Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on Predictability*, volume 1, pages 1–18. ECMWF, 1996.

[24] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5.

[25] Yongquan Qu, Juan Nathaniel, Shuolin Li, and Pierre Gentine. Deep generative data assimilation in multimodal setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 449–459, 2024.

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

[27] Louis Serrano, Thomas X Wang, Etienne Le Naour, Jean-Noël Vittaut, and Patrick Gallinari. Aroma: Preserving spatial structure for latent pde modeling with local neural fields. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024.

[28] Phillip Si and Peng Chen. Latent-ensf: A latent ensemble score filter for high-dimensional data assimilation with sparse observation data. In *The Thirteenth International Conference on Learning Representations*, 2025.

[29] Ashutosh Singh, Ricardo Augusto Borsoi, Deniz Erdogmus, and Tales Imbiriba. Learning semilinear neural operators: A unified recursive framework for prediction and data assimilation. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Adam Smith, Neal Lott, and Russ Vose. The integrated surface database: Recent developments and partnerships. *Bulletin of the American Meteorological Society*, 92(6):704–708, 2011.

[31] Andrew M. Stuart. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, 2010. doi: 10.1017/S0962492910000061.

[32] Jing-An Sun, Hang Fan, Junchao Gong, Ben Fei, Kun Chen, Fenghua Ling, Wenlong Zhang, Wanghan Xu, Li Yan, Pierre Gentine, and Lei Bai. Align-da: Align score-based atmospheric data assimilation with multiple preferences. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[33] Michael K Tippett, Jeffrey L Anderson, Craig H Bishop, Thomas M Hamill, and Jeffrey S Whitaker. Ensemble square root filters. *Monthly weather review*, 131(7):1485–1490, 2003.

[34] Yannick Tr'emolet. Accounting for an imperfect model in 4d-var. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 132(621):2483–2504, 2006.

[35] Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, Alexandre Adam, Jarrid Rector-Brooks, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Amortizing intractable inference in diffusion models for vision, language, and control. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024.

[36] Sifan Wang, Zhikai Wu, David van Dijk, and Lu Lu. Geofunflow: Geometric function flow matching for inverse operator learning over complex geometries. *arXiv preprint arXiv:2509.24117*, 2025.

[37] Tian Wang and Chuang Wang. Latent neural operator for solving forward and inverse pde problems. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024.

[38] Pengpeng Xiao, Phillip Si, and Peng Chen. Ld-ensf: Synergizing latent dynamics with ensemble score filters for fast data assimilation with sparse observations. In *The Fourteenth International Conference on Learning Representations*, 2026.

[39] Yi Xiao, Hang Fan, Kun Chen, Ye Cao, Ben Fei, Wei Xue, and Lei Bai. Lora-envar: Parameter-efficient hybrid ensemble variational assimilation for weather forecasting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[40] Yi Xiao, Qilong Jia, Kun Chen, Lei Bai, and Wei Xue. Vae-var: Variational autoencoder-enhanced variational methods for data assimilation in meteorology. In *The Thirteenth International Conference on Learning Representations*, 2025.

[41] Xingyu Xu and Yuejie Chi. Provably robust score-based diffusion posterior sampling for plug-and-play image reconstruction. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024.

[42] Yiming Yang, Xiaoyuan Cheng, Daniel Giles, Sibo Cheng, Yi He, Xiao Xue, Boli Chen, and Yukun Hu. Tensor-var: Efficient four-dimensional variational data assimilation. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 70649–70679, 2025.

[43] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, 2017.

A Additional Method Details

A.1 Finite-Rank Realization and Relation to Classical Control-Space DA

Section 3 describes ObsConDA in terms of abstract correction families, operators, and priors. The practical solver, however, works with their sampled finite-rank realizations on the latent grid $\Xi = \{\xi_m\}_{m=1}^M$. This subsection makes that transition explicit by rewriting the same objects in the solver's finite-dimensional form. The correspondence is:

$$\mathcal{F}(\mathbf{z}_b) \to B_\eta(\mathbf{z}_b), \qquad \widehat{\mathcal{B}}(\mathbf{z}_b) \to \big(B_\eta(\mathbf{z}_b), \Lambda_b(\mathbf{z}_b)\big), \qquad \mathcal{O}_\psi(\mathcal{G}) \to O_\psi(\mathcal{G}),$$
$$\widetilde{\mathcal{F}}(\mathbf{z}_b, \mathcal{G}) \to \widetilde{B}_{\mathcal{G}}, \qquad \text{control prior} \to P_{\mathcal{G}}.$$

Here the calligraphic symbols denote the abstract objects from Section 3, while $B_\eta(\mathbf{z}_b)$, $O_\psi(\mathcal{G})$, $\widetilde{B}_{\mathcal{G}}$, and $P_{\mathcal{G}}$ are their sampled finite-rank realizations on the latent grid. In what follows, each latent field on Ξ is flattened into $\mathbb{R}^M$ with $M = C_z H_z W_z$.

Background-conditioned objects. The first pair of finite-rank objects comes entirely from the current background state. A background-conditioned control-family generator g_η outputs the sampled control matrix

$$B_\eta(\mathbf{z}_b) = [\phi_k(\xi_m; \mathbf{z}_b)]_{m,k} \in \mathbb{R}^{M \times K}. \tag{9}$$

In parallel, a small background spectral head s_η outputs nonnegative control variances

$$\Lambda_b(\mathbf{z}_b) = \mathrm{Diag}(\mathrm{softplus}(s_\eta(\mathbf{z}_b))) \in \mathbb{R}^{K \times K}, \tag{10}$$

These variances induce the operational low-rank prior surrogate

$$\widehat{\mathcal{B}}(\mathbf{z}_b) = B_\eta(\mathbf{z}_b)\Lambda_b(\mathbf{z}_b)B_\eta(\mathbf{z}_b)^\top + \epsilon_{\mathcal{B}} I. \tag{11}$$

This is the operational object behind the "low-rank surrogate of local prior geometry" in the main text. It summarizes which latent directions remain plausible around the current background before the current observation layout is applied, and during support-aligned pretraining it supplies the prior geometry used to form prior-weighted observation backprojections. We never form $\widehat{\mathcal{B}}(\mathbf{z}_b)$ densely; all uses are matrix-free through the factorization in Eq. (11).

Geometry-conditioned objects. The second pair of finite-rank objects comes from the current observation geometry. We first encode the geometry with a permutation-invariant DeepSets encoder [43]

$$q_{\mathcal{G}} = \frac{1}{N}\sum_{i=1}^{N} \rho_\psi(p_i, \tau_i, c_i) \in \mathbb{R}^{d_q}, \tag{12}$$

where ρ_ψ is a shared multilayer perceptron over geometry tuples. The resulting geometry context $q_{\mathcal{G}}$ is then sent to two lightweight heads. The first head produces the finite-dimensional admissibility operator

$$O_\psi(\mathcal{G}) = \mathrm{Diag}(\sigma(a_\psi(q_{\mathcal{G}}))) \left(I + \frac{1}{\sqrt{r}} U_\psi(q_{\mathcal{G}}) V_\psi(q_{\mathcal{G}})^\top \right) \in \mathbb{R}^{K\times K}, \tag{13}$$

where σ is a sigmoid gate in $(0,1)$, and $U_\psi(q_{\mathcal{G}}), V_\psi(q_{\mathcal{G}}) \in \mathbb{R}^{K\times r}$ provide a low-rank coupling term. The diagonal gate suppresses unsupported directions, and the low-rank factor mixes the survivors. Applying this operator to the sampled candidate family yields

$$\widetilde{B}_{\mathcal{G}} = B_\eta(\mathbf{z}_b) O_\psi(\mathcal{G}), \tag{14}$$

which is the finite-rank realization of the admissible family used by the practical solver. Here $B_\eta(\mathbf{z}_b)$ and $\widehat{\mathcal{B}}(\mathbf{z}_b)$ carry the background dependence, while $O_\psi(\mathcal{G})$ only restricts and mixes that background-conditioned candidate family according to the current observation layout.

A second head uses the same geometry context to produce the control prior. We use the PSD parameterization

$$P_{\mathcal{G}} = \mathrm{Diag}(\mathrm{softplus}(p_\psi(q_{\mathcal{G}}))) + L_\psi(q_{\mathcal{G}}) L_\psi(q_{\mathcal{G}})^\top + \epsilon_P I, \tag{15}$$

with $L_\psi(q_{\mathcal{G}}) \in \mathbb{R}^{K\times r_P}$ and $\epsilon_P > 0$. This is the same geometry-aware control prior from Section 3; inside the solver it appears as the quadratic regularizer $\frac{1}{2}\boldsymbol{\alpha}^\top P_{\mathcal{G}} \boldsymbol{\alpha}$. The parameterization ensures $P_{\mathcal{G}}$ is positive definite while still allowing geometry-dependent anisotropy and cross-control coupling. Unlike $\widehat{\mathcal{B}}(\mathbf{z}_b)$, which summarizes background-local plausibility in latent space, $P_{\mathcal{G}}$ acts only after restriction, as a control-space regularizer over the surviving coefficients.

Three simple consequences. The next three propositions isolate the three roles of this finite-rank realization that are used most often in the paper: its limiting-case connection to classical reduced-control DA, its ability to shrink the effective control dimension under restrictive geometry, and its connection to the prior-weighted observation span that motivates the learned support surrogate.

Proposition 1 (Recovery of classical reduced-control variational DA)**.** *If the latent workspace is identified with the physical state space so that $D = I$, the control functions ϕ_k are fixed basis functions independent of $\mathbf{z}_b$, $O_\psi(\mathcal{G}) = I$, the projector satisfies $P_\omega = I$, and the physical regularizer is disabled, then minimizing the control-space objective in Eq.* (7) *reduces to a standard classical reduced-control variational DA objective over the coefficients $\boldsymbol{\alpha}$.*

This first proposition records the limiting case in which ObsConDA stops adapting its support to the current cycle and falls back to a classical reduced-control variational formulation.

Proposition 2 (Geometry-restricted control dimension)**.** *The admissible sampled control family satisfies*

$$\dim \mathrm{span}(\widetilde{B}_{\mathcal{G}}) \le \mathrm{rank}\big(O_\psi(\mathcal{G})\big) \le K. \tag{16}$$

Hence sparse or weakly informative observation layouts can explicitly contract the effective control dimension.

The second proposition makes the support-first viewpoint operational: once geometry suppresses directions through $O_\psi(\mathcal{G})$, the solver truly works in a smaller effective control subspace rather than merely carrying around unused coordinates.

Proposition 3 (Local MAP increment lies in a prior-weighted observation span)**.** *Consider the local linear-Gaussian objective in Eq.* (6) *with the physical regularizer disabled. Then the corresponding MAP increment satisfies*

$$\delta\mathbf{z}^\star = \mathcal{B}\mathcal{J}_\mathcal{G}^\top \boldsymbol{\lambda}^\star \tag{17}$$

for some coefficient vector $\boldsymbol{\lambda}^\star$. Hence $\delta\mathbf{z}^\star$ belongs to the span of prior-weighted observation backprojections $\mathrm{span}(\mathcal{B}\mathcal{J}_\mathcal{G}^\top)$.

The final proposition ties the learned support surrogate back to the local inverse problem. Under the same local model, the MAP increment lies in the same prior-shaped observation span from which the backprojected innovation directions are constructed. In practice, ObsConDA instantiates this span with the background-conditioned surrogate $\widehat{\mathcal{B}}(\mathbf{z}_b)$ from Eq. (11), so the support target and the learned control family are tied to the same low-rank prior geometry.

A.2 Control-Space Laplace Approximation and Complexity

This subsection records the two most important solver-side consequences of working in control coordinates: local posterior uncertainty can be approximated in the same low-dimensional space, and the resulting second-order algebra scales with K rather than with the ambient latent dimension M.

Around the final iterate $\boldsymbol{\alpha}^\star$, a Laplace approximation yields

$$\Sigma_{\text{post}} \approx \left(J_\star^\top R^{-1} J_\star + P_\mathcal{G} + \lambda_{\text{phys}} H_{\text{phys}} + \epsilon_I I\right)^{-1}, \tag{18}$$

where $J_\star$ is the control-space Jacobian at $\boldsymbol{\alpha}^\star$, H_{phys} is an optional Gauss-Newton approximation to the Hessian of the physical regularizer, and $\epsilon_I > 0$ stabilizes the inverse. We then draw control samples

$$\boldsymbol{\alpha}^{(s)} = \boldsymbol{\alpha}^\star + L_{\text{post}}\boldsymbol{\xi}^{(s)}, \qquad \boldsymbol{\xi}^{(s)} \sim \mathcal{N}(\mathbf{0}, I), \tag{19}$$

where $L_{\text{post}} L_{\text{post}}^\top = \Sigma_{\text{post}}$. That is, uncertainty is approximated locally around the converged control solution rather than over the full latent state directly.

Proposition 4 (Control-space linear-algebra complexity)**.** *Let $K \ll M$ be the control dimension and let $J_t \in \mathbb{R}^{N\times K}$ denote the already-evaluated control-space Jacobian. One control-space Gauss-Newton step requires $\mathcal{O}(NK^2 + K^3)$ operations to form and solve the normal equations, plus $\mathcal{O}(MK)$ to apply the sampled admissible control family. These counts describe the linear-algebra cost once Jacobian evaluation through the projected decoded observation map has been carried out. A corresponding dense latent-space second-order update would scale with M instead of K for the same normal-equation algebra, requiring $\mathcal{O}(NM^2 + M^3)$ operations.*

This proposition is meant to clarify where tractability comes from in ObsConDA: the expensive forward map still exists, but the normal-equation solve itself remains low-dimensional once the admissible family has been instantiated.

A.3 Latent Workspace, Projector, and Training Objectives

This subsection spells out the three training stages referenced in Section 3. The goal is to keep the roles of the modules separate while still training the final system end to end:

1. latent-workspace pretraining, which also includes projector pretraining;
2. support-aligned pretraining of the control-family generator together with the geometry encoder and admissibility operator;
3. end-to-end training, which introduces the geometry-aware control prior and unrolled solver while finetuning the earlier modules.

This subsection states what is optimized at each stage. The released supplemental configuration files state the exact loss weights, optimizer settings, solver step counts, latent dimensions, and schedule choices used for the reported runs, and we treat those configs as the authoritative hyperparameter record.

Stage 1: latent workspace and projector. The first stage has two jobs: it learns a latent workspace whose decoded fields remain physically plausible, and it trains the projector that will later keep solver proposals close to that workspace. We first train the latent workspace with

$$\begin{aligned}\mathcal{L}_{\text{manifold}} = \lambda_{\text{rec}} \|D(E(\mathbf{x})) - \mathbf{x}\|_1 \\ + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \lambda_{\text{cv}}\mathcal{L}_{\text{cv}} + \lambda_{\text{fcst}}\mathcal{L}_{\text{fcst}},\end{aligned} \tag{20}$$

where $\widehat{\mathbf{x}} = D(E(\mathbf{x}))$ is the reconstructed field. The reconstruction term anchors pointwise fidelity, while the three auxiliary terms preserve multiscale structure, cross-variable consistency, and short-horizon forecast utility. Concretely,

$$\mathcal{L}_{\text{spec}} = \sum_{c=1}^{C} \left\| \log\big(1 + |\,\mathrm{FFT}(\widehat{\mathbf{x}}_c)|\big) - \log\big(1 + |\,\mathrm{FFT}(\mathbf{x}_c)|\big) \right\|_1, \tag{21}$$

$$\mathcal{L}_{\text{cv}} = \left\| \mathcal{C}_{\text{diag}}(\widehat{\mathbf{x}}) - \mathcal{C}_{\text{diag}}(\mathbf{x}) \right\|_1, \tag{22}$$

$$\mathcal{L}_{\text{fcst}} = \left\| \mathcal{M}_{\Delta}(\widehat{\mathbf{x}}) - \mathcal{M}_{\Delta}(\mathbf{x}) \right\|_1, \tag{23}$$

where FFT is a channelwise 2D Fourier transform and

$$\mathcal{C}_{\text{diag}}(\mathbf{x}) = \mathrm{concat}\left(\{\partial_x \mathbf{x}_c, \partial_y \mathbf{x}_c, \Delta \mathbf{x}_c\}_{c=1}^{C}, \{\partial_x \mathbf{x}_c - \partial_y \mathbf{x}_{c'}\}_{(c,c')\in\mathcal{P}} \right) \tag{24}$$

is a fixed finite-difference diagnostic stack, with $\Delta \mathbf{x}_c$ denoting the spatial Laplacian of channel c and $\mathcal{P} = \{(c, c') : 1 \le c < c' \le C\}$ in our experiments. When explicit horizontal velocity channels are available, we append the usual divergence and vorticity diagnostics to the same stack. The forecast surrogate $\mathcal{M}_\Delta$ is a frozen one-step predictor trained separately on consecutive analysis pairs with the same backbone family as the autoencoder decoder and then kept fixed during ObsConDA training.

After the latent workspace has been shaped by Eq. (20), we train the background-conditioned projector P_ω as a latent denoising proximal map. Given a paired background-analysis cycle, let $\mathbf{z}_b = E(\mathbf{x}_b)$ be the background latent state and let $\mathbf{z} = E(\mathbf{x}_a)$ be the corresponding clean latent target. With perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{proj}}^2 I)$, we optimize

$$\mathcal{L}_{\text{proj}} = \|P_\omega(\mathbf{z} + \boldsymbol{\epsilon}, \mathbf{z}_b) - \mathbf{z}\|_2^2 + \lambda_{\text{fix}} \|P_\omega(\mathbf{z}, \mathbf{z}_b) - \mathbf{z}\|_2^2 . \tag{25}$$

The first term teaches the projector to denoise a perturbed latent proposal back toward the clean latent target, while the second fixes clean latents as approximate fixed points. At inference time, ObsConDA applies the same projector inside the observation term through $\bar{\mathbf{z}}(\boldsymbol{\alpha}) = P_\omega(\widetilde{\mathbf{z}}(\boldsymbol{\alpha}), \mathbf{z}_b)$, so projection is part of the optimized forward map rather than a post-hoc correction.

Stage 2: support-aligned family pretraining. During support-aligned control-family pretraining, we keep the latent autoencoder and projector fixed, train the control-family generator jointly with the geometry encoder and admissibility operator, and do not yet use the geometry-aware control prior inside a solver objective. At this stage $P_{\mathcal{G}}$ is not yet trained; the goal is only to learn a candidate family and its geometry-conditioned restriction. This stage uses one target from the true latent analysis increment and one target from the local inverse problem. Let

$$\Delta \mathbf{z}^\star = \mathbf{z}_a - \mathbf{z}_b$$

be the true latent analysis increment. This target tells the learned family which directions are needed to reconstruct realistic latent updates. The second target is the prior-weighted innovation backprojection

$$\mathbf{g}_{\mathcal{G}} = \widehat{\mathcal{B}}(\mathbf{z}_b) \mathcal{J}_{\mathcal{G}}^\top R^{-1} \mathbf{r}_{\mathcal{G}} \tag{26}$$

which captures observation-visible directions at the current cycle. Given the sampled admissible family $\widetilde{B}_{\mathcal{G}}$, we next construct two auxiliary quantities. The first is a ridge control code [18]

$$\boldsymbol{\alpha}^\dagger = \arg\min_{\boldsymbol{\alpha}} \left\| \Delta \mathbf{z}^\star - \widetilde{B}_{\mathcal{G}} \boldsymbol{\alpha} \right\|_2^2 + \lambda_\alpha \|\boldsymbol{\alpha}\|_2^2, \tag{27}$$

which serves only as an auxiliary support-alignment code for this pretraining stage and is distinct from the solver output $\boldsymbol{\alpha}^\star$ used in the final end-to-end stage. The second is the projection matrix

$$\Pi_{\mathcal{G}} = \widetilde{B}_{\mathcal{G}} \left(\widetilde{B}_{\mathcal{G}}^\top \widetilde{B}_{\mathcal{G}} + \epsilon_B I \right)^{-1} \widetilde{B}_{\mathcal{G}}^\top . \tag{28}$$

This is the ridge-stabilized projector onto the span of the current sampled admissible family. Intuitively, $\boldsymbol{\alpha}^\dagger$ measures how well the current family reconstructs the true latent increment, while $\Pi_{\mathcal{G}}$ measures whether the observation-visible directions already lie inside that family. We then optimize the support-aligned family loss

$$\mathcal{L}_{\text{fam}} = \lambda_{\text{inc}} \left\| \Delta \mathbf{z}^\star - \widetilde{B}_{\mathcal{G}} \boldsymbol{\alpha}^\dagger \right\|_2^2 + \lambda_{\text{supp}} \left\| (I - \Pi_{\mathcal{G}}) \mathbf{g}_{\mathcal{G}} \right\|_2^2 + \lambda_{\text{div}} \mathcal{L}_{\text{div}}, \tag{29}$$

where $\mathcal{L}_{\text{div}}$ encourages diversity among the learned control functions. Concretely, letting $\bar{B}_\eta(\mathbf{z}_b)$ denote the column-normalized version of $B_\eta(\mathbf{z}_b)$, we use

$$\mathcal{L}_{\text{div}} = \left\| \bar{B}_\eta(\mathbf{z}_b)^\top \bar{B}_\eta(\mathbf{z}_b) - I \right\|_F^2 . \tag{30}$$

The first term teaches the family to reconstruct realistic latent analysis increments, the second forces observation-visible directions to lie inside the learned admissible family, and the third discourages redundant control directions. In practice, the backprojection in $\mathbf{g}_{\mathcal{G}}$ is applied matrix-free as $B_\eta(\Lambda_b(B_\eta^\top \mathbf{u})) + \epsilon_{\mathcal{B}} \mathbf{u}$ with $\mathbf{u} = \mathcal{J}_{\mathcal{G}}^\top R^{-1} \mathbf{r}_{\mathcal{G}}$.

Stage 3: end-to-end solver training. For end-to-end training, we initialize from the pretrained latent workspace, projector, control-family generator, geometry encoder, and admissibility operator; we then introduce the geometry-aware control prior and unrolled solver, and finetune all trainable modules jointly. We randomize observation subsets and layouts so that the same background trajectory is seen under multiple geometries. Let $\boldsymbol{\alpha}^\star$ denote the output of the T-step unrolled control-space solver from Section 3. Let $\widehat{\widetilde{\mathbf{z}}}_a = \mathbf{z}_b + \widetilde{B}_{\mathcal{G}} \boldsymbol{\alpha}^\star$ be the final admissible proposal, let $\widehat{\mathbf{z}}_a = P_\omega(\widehat{\widetilde{\mathbf{z}}}_a, \mathbf{z}_b)$ be the projected latent analysis used by the solver, and let $\widehat{\mathbf{x}}_a = D(\widehat{\mathbf{z}}_a)$ be the decoded analysis output. We optimize

$$\mathcal{L}_{\text{e2e}} = \lambda_a \mathcal{L}_{\text{analysis}} + \lambda_o \mathcal{L}_{\text{obs}} + \lambda_p \mathcal{L}_{\text{phys}} + \lambda_f \mathcal{L}_{\text{fcst}} + \lambda_c \mathcal{L}_{\text{calib}} + \lambda_{\text{div}} \mathcal{L}_{\text{div}}, \tag{31}$$

with

$$\mathcal{L}_{\text{analysis}} = \| \widehat{\mathbf{x}}_a - \mathbf{x}_a \|_1, \tag{32}$$

$$\mathcal{L}_{\text{obs}} = \| H_{\mathcal{G}}(\widehat{\mathbf{x}}_a) - \mathbf{v} \|_{R^{-1}}^2 , \tag{33}$$

$$\mathcal{L}_{\text{phys}} = \mathcal{R}_{\text{phys}}(\widehat{\widetilde{\mathbf{z}}}_a, \mathbf{x}_b), \tag{34}$$

$$\mathcal{L}_{\text{fcst}} = \| \mathcal{M}_\Delta(\widehat{\mathbf{x}}_a) - \mathcal{M}_\Delta(\mathbf{x}_a) \|_1, \tag{35}$$

$$\mathcal{L}_{\text{calib}} = -\log \mathcal{N}(\boldsymbol{\alpha}^\dagger; \boldsymbol{\alpha}^\star, \Sigma_{\text{post}}), \tag{36}$$

where $\mathcal{M}_\Delta$ is a frozen short-horizon forecast surrogate used only during training, and Σ_{post} is the control-space Laplace covariance from Eq. (18). If posterior uncertainty is not trained, we simply set $\lambda_c = 0$. The calibration term is optional and intentionally auxiliary. It reuses the code $\boldsymbol{\alpha}^\dagger$ from the support-aligned stage as a stable control-space surrogate derived from the true latent increment under the current admissible family, but it does not replace the solver output $\boldsymbol{\alpha}^\star$ as the posterior mean, nor does it treat $\boldsymbol{\alpha}^\dagger$ as an exact posterior sample or exact MAP target. Instead, it only nudges the local Laplace covariance around $\boldsymbol{\alpha}^\star$ to assign reasonable mass to a surrogate activation that reconstructs the true latent increment well. The deterministic analysis and observation-fit losses remain the primary training signals; calibration quality must still be validated empirically through the reliability and coverage experiments in the main paper.

A.4 ObsConDA Inference Algorithm

Algorithm 1 summarizes the final test-time inference pipeline after the background-conditioned candidate family, geometry-conditioned restriction, and projector have all been trained.

B Additional Experimental Results

Appendix B defines the reported metrics, breaks Track A down by variable, adds a Track A station-budget check, adds Track B qualitative cases, reports a separate marginal predictive comparison, and closes with runtime comparisons.

Algorithm 1 ObsConDA Inference in Control Coordinates

Require: Background forecast $\mathbf{x}_b$, observations $\mathcal{Y}$
$\mathbf{z}_b \leftarrow E(\mathbf{x}_b)$
$(\mathbf{v}, \mathcal{G}) \leftarrow \textsc{SplitObservations}(\mathcal{Y})$
Build the sampled admissible correction family $\widetilde{B}_{\mathcal{G}}$
Form the geometry-aware control prior $P_{\mathcal{G}}$
$\boldsymbol{\alpha}_0 \leftarrow \mathbf{0}$
for $t = 0, 1, \ldots, T-1$ **do**
    $\widetilde{\mathbf{z}}_t \leftarrow \mathbf{z}_b + \widetilde{B}_{\mathcal{G}} \boldsymbol{\alpha}_t$
    $\bar{\mathbf{z}}_t \leftarrow P_\omega(\widetilde{\mathbf{z}}_t, \mathbf{z}_b)$
    $\mathbf{r}_t \leftarrow \mathbf{v} - H_{\mathcal{G}}(D(\bar{\mathbf{z}}_t))$
    Form the projected control-space Jacobian J_t
    Solve Eq. (8) for $\Delta\boldsymbol{\alpha}_t$
    $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t + \beta_t \Delta\boldsymbol{\alpha}_t$
end for
$\widetilde{\mathbf{z}}_a \leftarrow \mathbf{z}_b + \widetilde{B}_{\mathcal{G}} \boldsymbol{\alpha}_T$
$\widehat{\mathbf{z}}_a \leftarrow P_\omega(\widetilde{\mathbf{z}}_a, \mathbf{z}_b)$
return $\widehat{\mathbf{x}}_a = D(\widehat{\mathbf{z}}_a)$

B.1 Metric Definitions

Core notation. Across Tracks A–C, $y_{v,i}$ and $\hat{y}_{v,i}$ denote the target and prediction at an observation or held-out location i, while $x_v(\xi)$ and $\hat{x}_v(\xi)$ denote the target and prediction at grid point $\xi \in \Omega$. The definitions below are written in observation space; the full-field and state-space versions replace y with x and average over the relevant grid points or state coordinates. When a method provides a Gaussian predictive distribution with mean and standard deviation (μ, σ), we report:

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{37}$$

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{38}$$

$$\mathrm{NLL}(y; \mu, \sigma) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}\log(2\pi\sigma_i^2) + \frac{(y_i - \mu_i)^2}{2\sigma_i^2}\right]. \tag{39}$$

For the Gaussian continuous ranked probability score (CRPS) [13, 16], we define $z_i = (y_i - \mu_i)/\sigma_i$ and use

$$\mathrm{CRPS}(y; \mu, \sigma) = \frac{1}{N}\sum_{i=1}^{N}\sigma_i\left[z_i\big(2\Phi(z_i) - 1\big) + 2\phi(z_i) - \frac{1}{\sqrt{\pi}}\right], \tag{40}$$

where Φ and ϕ are the standard normal CDF and PDF. For nominal coverage $\alpha \in \{0.5, 0.8, 0.9\}$, let $z_\alpha = \Phi^{-1}((1+\alpha)/2)$ and define the empirical coverage and coverage error as

$$\widehat{c}_\alpha = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\{\mu_i - z_\alpha\sigma_i \le y_i \le \mu_i + z_\alpha\sigma_i\}, \tag{41}$$

$$\mathrm{CovErr}_\alpha = |\widehat{c}_\alpha - \alpha|. \tag{42}$$

Normalized aggregates. Track A also reports normalized aggregate metrics across $\mathcal{V} = \{u10, v10, t2m, msl\}$. If s_v denotes the benchmark scale for variable v and $\mathcal{H}_v$ the held-out station set for that variable, then

$$\text{Held-out NRMSE} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\sqrt{\frac{1}{|\mathcal{H}_v|}\sum_{i\in\mathcal{H}_v}\left(\frac{\hat{y}_{v,i} - y_{v,i}}{s_v}\right)^2}, \tag{43}$$

Table 5: Per-variable held-out Track A results on the real surface-station benchmark.

Method	$u10$				$v10$				$t2m$				msl			
	RMSE	MAE	NLL	CRPS	RMSE	MAE	NLL	CRPS	RMSE	MAE	NLL	CRPS	RMSE	MAE	NLL	CRPS
Background	1.727	1.284	2.006	0.960	1.783	1.354	2.017	0.990	1.883	1.289	2.072	0.999	102.391	59.766	6.467	68.186
IDW/OI	1.602	1.187	1.919	0.869	1.665	1.222	1.979	0.898	1.857	1.268	2.136	0.953	92.152	51.290	6.091	48.477
3DVar	1.594	1.186	1.907	0.879	1.649	1.221	1.933	0.902	1.827	1.271	2.030	0.963	90.388	53.241	6.184	53.090
FNP	1.628	1.220	1.962	0.915	1.678	1.250	1.977	0.933	1.805	1.248	2.043	0.970	95.248	56.793	6.393	63.783
VAE-Var	1.627	1.220	1.961	0.914	1.676	1.248	1.976	0.932	1.798	1.243	2.040	0.967	95.568	56.808	6.395	63.919
ObsConDA	**1.514**	**1.139**	**1.905**	**0.863**	**1.546**	**1.147**	**1.915**	**0.872**	**1.706**	**1.203**	**1.996**	**0.935**	**90.120**	**51.180**	**6.075**	**48.300**

Table 6: Per-variable full-field Track A results against the ERA5 analysis target.

Method	$u10$		$v10$		$t2m$		msl	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Background	0.441	0.306	0.446	0.309	0.515	0.321	32.750	23.238
IDW/OI	0.438	0.303	0.444	0.306	0.507	0.314	32.320	22.980
3DVar	0.435	0.300	0.441	0.304	0.500	0.309	31.980	22.760
FNP	0.430	0.295	0.435	0.298	0.489	0.295	30.980	21.980
VAE-Var	0.426	0.292	0.431	0.295	0.480	0.289	30.420	21.520
ObsConDA	**0.374**	**0.245**	**0.374**	**0.242**	**0.408**	**0.226**	**27.579**	**18.496**

and the same normalization defines held-out NCRPS and full-field NRMSE. Track B uses the same per-variable metrics but averages them in two ways: surface metrics average over variables and then over all four geometry families, while geometry-OOD metrics use the same aggregation restricted to the shifted families {random irregular, block-missing, mixed density}.

Interval summaries. We also report an average interval width,

$$\mathrm{Width}_\alpha = \frac{1}{N}\sum_{i=1}^{N} 2z_\alpha\sigma_i,$$

and a reliability-slope deviation. In Table 7, the reported width is $\mathrm{Width}_{0.9}$, i.e., the average width of the nominal 90% interval. Writing $q \in \{0.1, 0.2, \ldots, 0.9\}$ for the nominal coverage grid and $\widehat{c}_q$ for the corresponding empirical coverage, we fit the reliability line $\widehat{c}_q \approx sq + b$ and report the slope deviation from the ideal value 1, namely $|1 - s|$; this follows the usual calibration-versus-sharpness viewpoint in probabilistic forecast evaluation [14].

Synthetic-system metrics. For Track C, 'state RMSE' and 'state NLL' apply the same formulas to the full state vector of each synthetic system. The 2D PDE spectrum diagnostic is the normalized mean absolute discrepancy between Fourier amplitudes,

$$\mathrm{SpectrumErr}(x, \hat{x}) = \frac{\mathrm{mean}\left(\left|\,|\mathcal{F}(\hat{x})| - |\mathcal{F}(x)|\,\right|\right)}{\mathrm{mean}\left(|\mathcal{F}(x)|\right)}, \tag{44}$$

where $\mathcal{F}$ denotes the 2D real FFT used in the evaluation code.

B.2 Track A Detailed Results

Per-variable results. Tables 5 and 6 unpack Track A by variable at held-out stations and over the full ERA5 analysis field.

Held-out results. Across all four variables, ObsConDA is best in every held-out RMSE, MAE, NLL, and CRPS column. The RMSE gains are broad: $u10$ falls from 1.594 for 3DVar to 1.514, $v10$ from 1.649 to 1.546, $t2m$ from 1.798 for VAE-Var to 1.706, and msl from 90.388 for 3DVar to 90.120. The same ordering carries through MAE, NLL, and CRPS, so the Track A station gains are not concentrated in one variable.

Full-field results. ObsConDA is again best in every RMSE and MAE column, lowering $u10$ RMSE from 0.426 for VAE-Var to 0.374, $v10$ from 0.431 to 0.374, $t2m$ from 0.480 to 0.408, and msl from 30.420 to 27.579. The MAE columns follow the same pattern, with the largest absolute reduction again on pressure, from 21.520 to 18.496. The field-space gain in Table 1 is broad across wind, temperature, and pressure rather than driven by one variable alone.

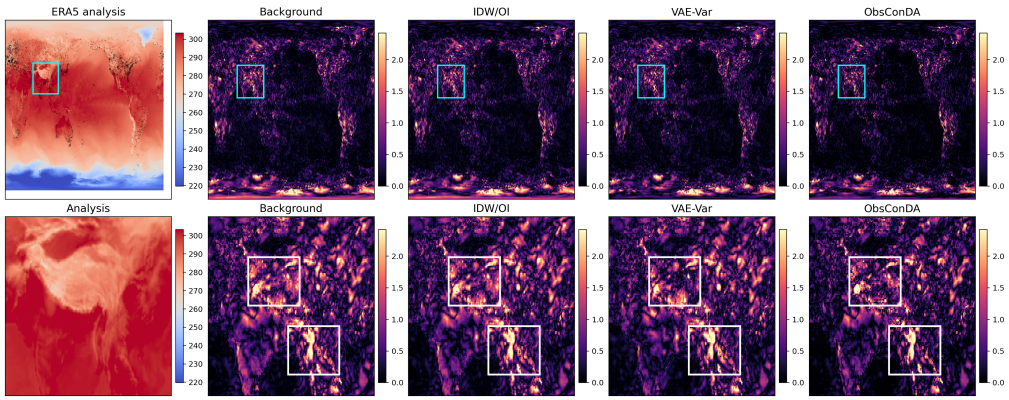

Figure 3: Track A qualitative case at 2024-09-13 06:00 UTC. Top: target $t2m$ with valid station locations and full-domain absolute-error maps. Bottom: one boxed high-error region where ObsConDA remains comparatively better.

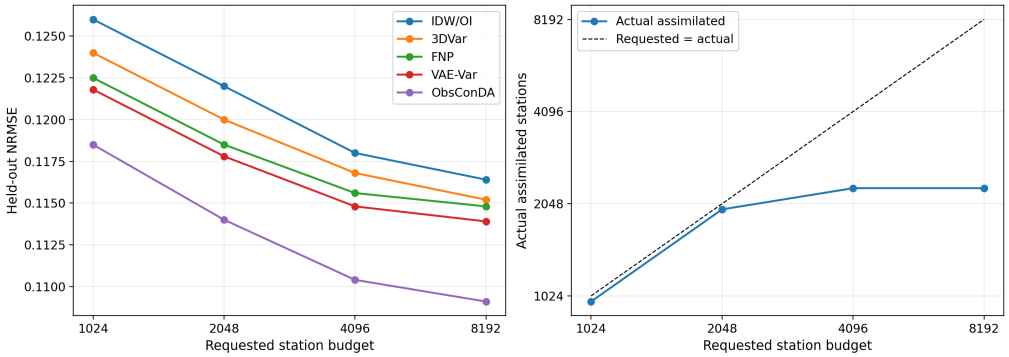

Figure 4: Track A station-budget scaling. The left panel shows held-out NRMSE against the requested station budget, and the right panel shows the corresponding number of assimilated stations.

Qualitative case. In this representative cycle, ObsConDA reduces the warm-side spatial bias that remains in the background and in the stronger interpolation-style baselines. The boxed region in the bottom row is still a hard part of the field for every method, but ObsConDA remains relatively better there as well. Its corrections also stay broad and meteorologically coherent under the same irregular station layout, which matches the full-field RMSE advantage reported in the main text.

Station-budget scaling. Figure 4 shows why the high-budget regime flattens. On the left, the classical baselines improve as the requested budget grows, and ObsConDA stays lowest among the methods shown, moving from 0.1185 at 1024 requested stations to 0.1091 at 8192. On the right, those same requested budgets correspond to only about 981, 1963, 2302, and 2302 assimilated stations after quality control and held-out reservation. The curve flattens because the available station pool is already exhausted.

B.3 Additional Track B Cases

The figures below show Track B qualitative cases under matched synthetic observations.

Matched-observation example. In this matched-observation cycle, ObsConDA suppresses the large coherent error lobes that remain under incomplete real-station geometry, even though all methods receive the same synthetic values under the same known noise model. The boxed region again marks a difficult part of the field where all methods still incur noticeable error, but ObsConDA remains relatively better than the alternatives. This matches the normalized NRMSE and NCRPS gains in Table 2.

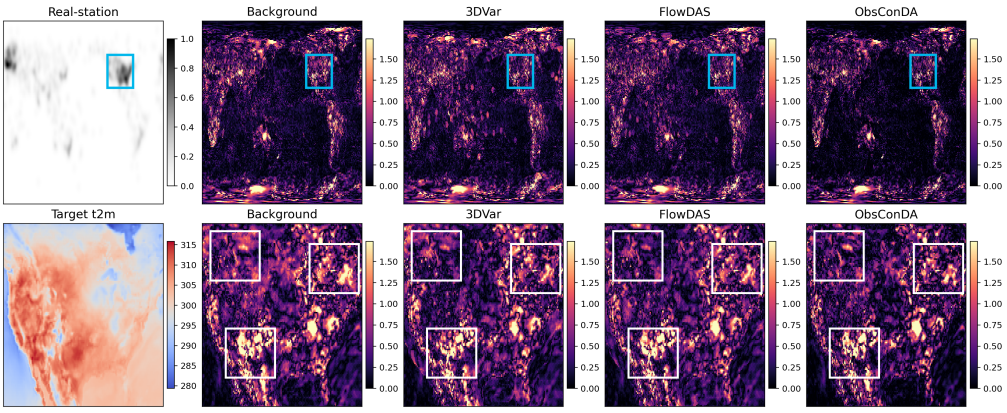

Figure 5: Track B matched-observation case at 2024-09-13 06:00 UTC. Top: observation density and full-domain absolute errors. Bottom: one boxed high-error region where ObsConDA remains comparatively better.

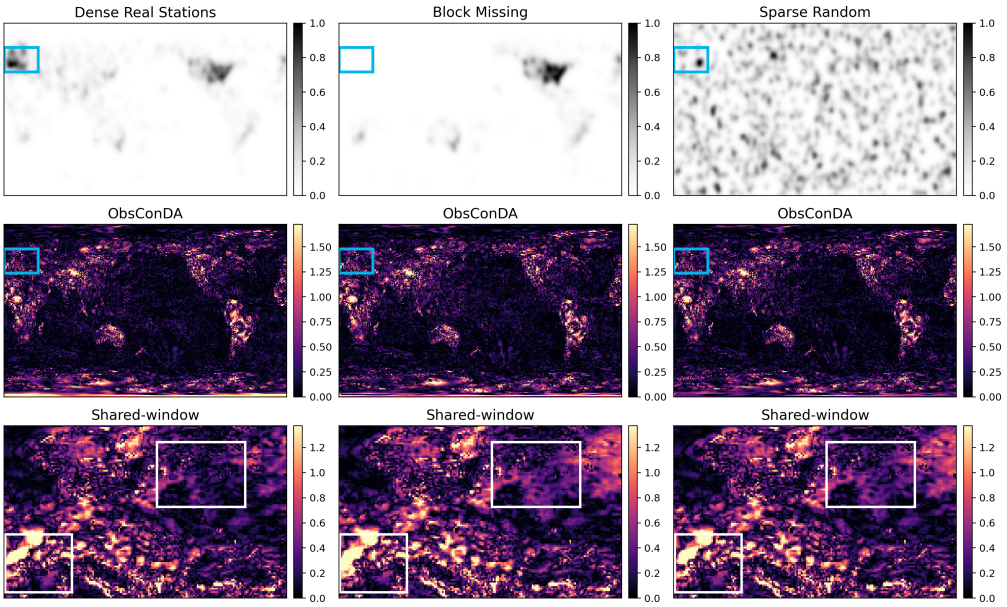

Figure 6: Track B geometry-dependent ObsConDA errors at 2024-09-13 06:00 UTC. Columns show different observation geometries; rows show observation density, full-domain absolute error, and the same local error window.

Geometry dependence. Figure 6 shows the same ObsConDA model under several observation geometries. The dense layout gives the cleanest result, while the error grows as the geometry becomes sparser or more structured. Even so, the errors remain compact instead of expanding into geometry-shaped artifacts after the layout changes from real-station placement to structured holes and mixed density. This matches the gap between full ObsConDA and the fixed-support variants in Table 3.

B.4 Calibration and Marginal Predictive Uncertainty

We also compare marginal predictive scores for the methods with usable uncertainty artifacts. FNP and VAE-Var use residual-calibrated scalar predictive standard deviations. 3DVar and the current FlowDAS adapter are omitted because they do not provide directly comparable predictive-distribution outputs in the present pipeline. Avg. Width is the average nominal 90% interval width, and Reliability Slope is the deviation of the fitted empirical-versus-nominal coverage slope from the ideal value 1.

Table 7: Marginal predictive calibration and probabilistic scores for methods with usable uncertainty artifacts.

Method	NLL ↓	CRPS ↓	50% Cov. Err. ↓	80% Cov. Err. ↓	90% Cov. Err. ↓	Avg. Width ↓	Reliability Slope ↓
FNP	1.773	4.122	0.112	0.058	0.034	25.631	0.032
VAE-Var	1.728	4.097	0.148	0.078	0.047	25.754	0.066
ObsConDA	**1.498**	**3.095**	**0.097**	**0.047**	**0.028**	**19.943**	**0.029**

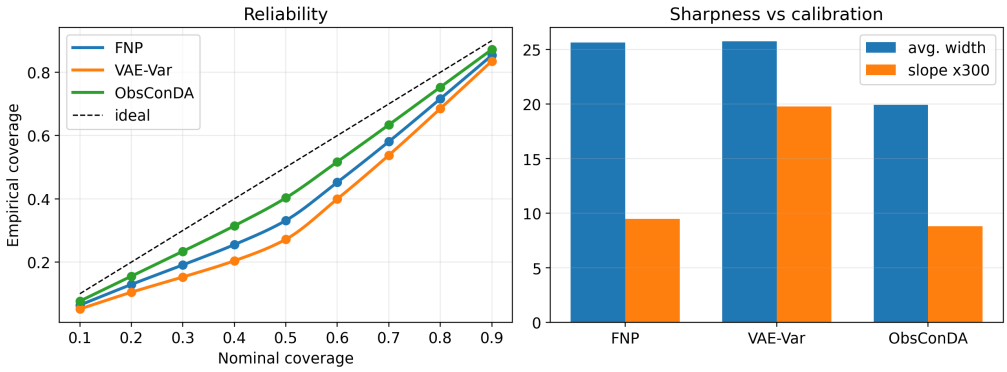

Figure 7: Reliability and interval-summary view for the marginal predictive comparison.

Calibration and scores. Table 7 favors ObsConDA in every row. Its NLL drops to 1.498 from 1.728 for VAE-Var and 1.773 for FNP, and its CRPS drops to 3.095 from 4.097 and 4.122. The intervals are also narrower at 19.94, compared with about 25.7 for the proxy baselines. The coverage rows follow the same ordering: ObsConDA reaches 0.097, 0.047, and 0.028 at the 50%, 80%, and 90% levels, compared with 0.112/0.058/0.034 for FNP and 0.148/0.078/0.047 for VAE-Var. Its reliability-slope deviation is likewise the smallest at 0.029. The sharper intervals also remain better calibrated in this comparison.

Reliability view. In the left panel of Figure 7, curves below the diagonal indicate under-coverage. ObsConDA stays closest to the ideal line throughout the displayed grid: at nominal coverages 0.5, 0.8, and 0.9, its empirical coverages are 0.403, 0.753, and 0.872, compared with 0.336/0.718/0.855 for FNP and 0.272/0.685/0.836 for VAE-Var. The right panel reports the same comparison through average 90% interval width and reliability-slope deviation. ObsConDA again gives the narrowest intervals at 19.94 and the smallest slope deviation at 0.029.

B.5 Runtime and Resource Comparison

Measurement setup. Table 8 reports runtime and GPU memory for the methods that were run in our paper pipeline. All measurements use a single RTX 4090. Time / cycle is the mean one-cycle inference time, and peak memory is one-cycle CUDA reserved memory. Because the methods do different amounts of work in one cycle, the work column lists the computation behind each reported time. The hardware, benchmark cycle, and output target are fixed, while each method is timed along its native one-cycle evaluation path. Peak memory is recorded from one-cycle runs under each method's evaluation configuration: a deterministic Track A full-surface pass for FNP, a native batch-4 export pass for VAE-Var, a 25-sample 3-step denoising pass for FlowDAS, and the formal Track A inference pass for ObsConDA.

Runtime summary. ObsConDA runs in the same range as FNP and VAE-Var: 0.084 s/cycle for ObsConDA, 0.084 for FNP, and 0.083 for VAE-Var. The localized 3DVar baseline is much slower at 831.068 s/cycle. ObsConDA also uses little reserved GPU memory at 0.047 GiB, slightly below FlowDAS (0.057 GiB) and well below VAE-Var (0.219 GiB). The work column lists the corresponding one-cycle computation.

Table 8: Runtime and GPU memory on a single RTX 4090.

Method	Time / cycle ↓	Peak memory ↓	Assimilation work	Posterior samples
3DVar	831.068	—	localized linear solve per held-out station	n/a
FNP	0.084	4.135 GiB	one deterministic function-space forward pass	n/a
VAE-Var	**0.083**	0.219 GiB	one deterministic encoder/decoder assimilation pass	n/a
FlowDAS	0.138	0.057 GiB	3 auto-regressive denoising steps	25 MC × 100 experiment samples
ObsConDA	0.084	**0.047 GiB**	four variable-wise rank-8 control solves plus projector	n/a

Table 9: Synthetic validation on Lorenz-96 and the 2D advection–diffusion benchmark.

Method	L96 RMSE ↓	L96 NLL ↓	L96 90% Cov. Err. ↓	2D PDE RMSE ↓	2D PDE Spectrum Err. ↓	2D PDE Geometry-OOD RMSE ↓
3DVar	1.35364	1.67790	0.04238	0.10952	0.50422	0.10947
ETKF	1.51456	1.87176	0.08965	0.14944	0.72031	0.14974
FlowDAS	1.36868	1.92067	0.08535	0.12486	0.39136	0.12624
ObsConDA w/ fixed support	1.29178	1.63901	**0.03926**	0.10533	0.47740	0.10519
ObsConDA w/o support loss	1.28926	1.65313	0.04004	0.08312	**0.28994**	0.08296
ObsConDA	**1.27008**	**1.62619**	0.03965	**0.08169**	0.31768	**0.08152**

C Validation on Synthetic Dynamical Systems

We also test ObsConDA on two standard synthetic DA settings. Lorenz-96 [23] is the main state-vector benchmark. It is a 40-dimensional system with periodic indices,

$$\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + F, \qquad j = 1, \dots, 40,$$

with forcing $F = 8$. In our benchmark we observe 25% of the coordinates in each cycle under fixed, random, or block masks with additive noise, and compare state estimation and probabilistic scores against a classical variational baseline (3DVar), an ensemble baseline (ETKF; 33), and learned baselines. The second benchmark is a two-dimensional advection–diffusion field-reconstruction problem,

$$\partial_t u = -a_x\, \partial_x u - a_y\, \partial_y u + \nu \Delta u + f(x, y), \qquad f(x, y) = 0.01 \sin(2\pi x) \sin(2\pi y),$$

with $(a_x, a_y) = (0.65, -0.35)$ and $\nu = 0.015$ on a periodic 16×16 grid. In this benchmark we observe 20% of the grid cells in each cycle under fixed, random, or block masks with additive noise, and evaluate both full-state accuracy and spectral fidelity. Lorenz-63 [22] is included as a smaller trajectory sanity check. FlowDAS is also reported on both benchmarks.

Synthetic results. On Lorenz-96, ObsConDA gives the lowest RMSE and NLL at 1.270 and 1.626, improving on 3DVar at 1.354/1.678, ETKF at 1.515/1.872, and FlowDAS at 1.369/1.921. The smallest 90% coverage error, 0.03926, comes from the fixed-support variant. On the 2D advection–diffusion benchmark, ObsConDA gives the lowest full-state RMSE and geometry-OOD RMSE at 0.08169 and 0.08152, while the lowest spectrum error, 0.28994, comes from the no-support-loss variant. The main reconstruction rows therefore favor full ObsConDA on both systems.

Advection–diffusion sample. In this sample, ObsConDA recovers the target field more cleanly than the background and FlowDAS, especially around the sharper local structures. The spectrum view points in the same direction: ObsConDA is closer to the truth on most of the displayed bins, while FlowDAS is only slightly closer at the lowest and highest shown bins. This matches Table 9, where the clearest gain appears in field reconstruction and the spectrum row stays competitive.

Lorenz-63 sanity check. The Lorenz-63 system follows

$$\dot{a} = \sigma(b - a), \qquad \dot{b} = a(\rho - c) - b, \qquad \dot{c} = ab - \beta c,$$

with the standard chaotic parameters $(\sigma, \rho, \beta) = (10, 28, 8/3)$. Figure 9 marks the short-horizon window, and Figure 10 shows the corresponding 3D trajectories. In that window, ObsConDA stays

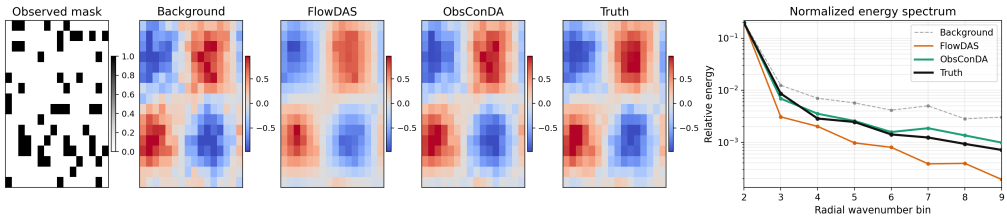

Figure 8: One 2D advection–diffusion sample under partial observation. The panels show the observation mask, the background, FlowDAS, ObsConDA, the target field, and the corresponding normalized energy spectra.

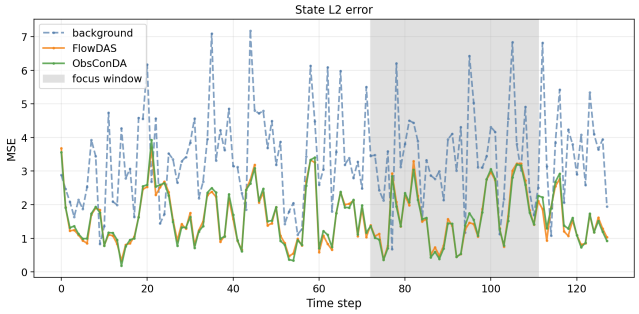

Figure 9: Lorenz-63 full-horizon state L_2 error, with the highlighted short-horizon window used below.

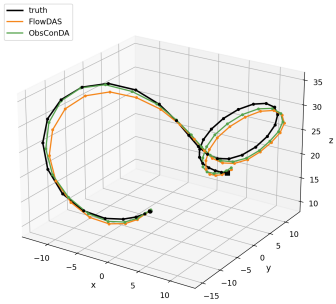

Figure 10: Lorenz-63 3D trajectory view in the highlighted short-horizon window.

on the same scale as the strongest baseline and follows the same attractor geometry without an early blow-up.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the core claim, method definition, and contribution list, and the results section evaluates those claims on real-weather, simulated-observation, and synthetic benchmarks; see the Abstract, Section 1, Section 3, and Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section that discusses latent-manifold dependence, likelihood simplifications, training complexity, and the need for stronger support-learning evidence; see Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [No]

Justification: The draft states several propositions and gives detailed finite-rank constructions in the appendix, but it does not yet include complete formal proofs for all stated theoretical results; see Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The supplemental material provides a reproduction-oriented package for the reported pipeline, including environment notes, major configurations, data-preparation and path conventions, executable commands, and an explicit statement of which baselines are artifact-first versus fully rerunnable in the current interface.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The submission is accompanied by anonymized supplemental code and reproduction materials for the main pipeline, including environment setup notes, scripts, and instructions needed to rerun the released artifact-backed experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the benchmark splits, variables, station budgets, and evaluation metrics, while the supplemental material provides the main configuration files, model settings, and run-path details needed to understand how the reported experiments were executed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The reported tables and figures do not currently include error bars, confidence intervals, or formal significance tests. The present results are point estimates over the benchmark cycles.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper states the main hardware budget and reports runtime and memory summaries in the appendix, while the supplemental material describes the practical execution setup and distinguishes full reruns from artifact-first baselines; see Section 3.4 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is a methodology and evaluation paper on weather data assimilation and does not involve deception, human subjects, or unsafe deployment procedures. We have reviewed the NeurIPS Code of Ethics and the current submission does not intentionally deviate from it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The current draft does not yet include a dedicated broader-impacts discussion. A final version should discuss both the potential benefits of improved environmental forecasting and the risks of misuse, overconfidence, or unequal operational access.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk foundation model, scraped web-scale dataset, or similarly misuse-prone asset. The artifacts are domain-specific weather-assimilation models and evaluation outputs.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper credits baseline methods and datasets in the text, but the current draft does not yet explicitly enumerate all relevant licenses and terms of use for external code, model, and data assets. This should be added before submission.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The current submission does not release a new public dataset, model package, or software asset with accompanying external documentation. It reports experiments and internal artifacts only.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing, participant studies, or other human-subject data collection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or research with human subjects, so IRB approval or an equivalent review was not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the proposed method, training pipeline, or experimental evaluation. Any LLM assistance, if used at all, was not an important or non-standard component of the research methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.