Control-Augmented Diffusion for Autoregressive Data Assimilation

Prakhar Srivastava
University of California, Irvine
prakhs2@uci.edu

Farrin Marouf Sofian
University of California, Irvine
fmaroufs@uci.edu

Francesco Immorlano
University of California, Irvine
fimmorla@uci.edu

Stephan Mandt
University of California, Irvine
mandt@uci.edu

Abstract

Data assimilation (DA) in chaotic spatiotemporal systems, such as turbulent partial differential equations (PDEs), is essential but computationally demanding, often requiring expensive adjoints, ensembles, or test-time optimization. We introduce an amortized framework that augments autoregressive diffusion models with learned feedback control. A pretrained diffusion model provides one-step forecasts, while a compact control network, trained offline, injects affine residuals into the DDIM denoising steps. These residuals gently nudge the sampler toward consistency with upcoming observations, preventing forecast drift during long observations gaps. With the proposed approach, DA at inference reduces to a single forward rollout with on-the-fly corrections, avoiding costly optimizations or ensembles. On chaotic Kolmogorov flow, our method yields improved long-horizon stability, substantial accuracy gains, and over $30\times$ faster runtime. To our knowledge, this is the first framework to integrate amortized assimilation directly into autoregressive diffusion models, opening a new direction for efficient learned control in high-dimensional PDE forecasting.

1 Introduction

Forecasting chaotic spatiotemporal dynamics is hard: small state errors grow rapidly, so open-loop rollouts drift from reality. Data assimilation (DA) mitigates this by combining a dynamical model with sparse, noisy observations to produce analyses that improve predictability [1, 2]. Operational staples such as 4D-Var and the Ensemble Kalman Filter (EnKF) [3, 4, 5] can be highly effective, but typically hinge on quasi-linear approximations and incur heavy adjoint/ensemble costs [6]. Diffusion models provide a flexible Bayesian alternative for reconstructing full states from partial data. Recent diffusion-based DA methods either (i) apply guidance only at inference [7, 8, 9] or (ii) train models to condition directly on observations [10, 11]. However, inference-only guidance can allow errors to accumulate between observation times, naïve conditioning can destabilize long autoregressive forecasts, and iterative denoising can make inference slow.

We introduce a diffusion-based DA framework that inserts a lightweight, learned controller into the generative dynamics. A pretrained diffusion forecaster defines the backbone transitions, while a control network injects affine residuals into each Denoising Diffusion Implicit Model (DDIM) step (Fig. 1), producing preview-aware corrections that steer toward upcoming observations without modifying the backbone. The controller is trained offline on synthetic assimilation scenarios, so test-time DA is a single causal, feed-forward rollout with on-the-fly corrections—avoiding per-instance optimization while improving stability, accuracy, and speed in chaotic PDE forecasting.

Accepted at the AI for Science workshop (NeurIPS 2025).

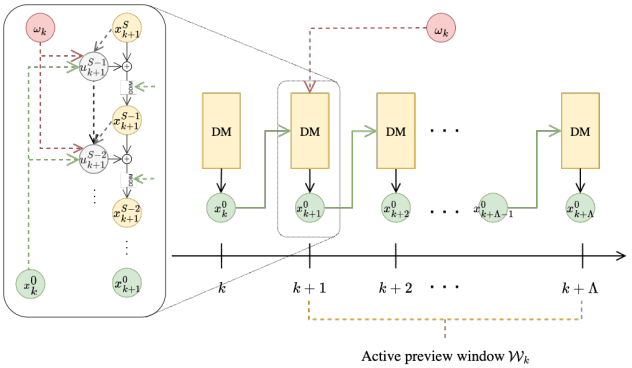

Figure 1: At step k, a pretrained diffusion backbone advances x_k to x_{k+1} via S DDIM denoising sub-steps, generating latents $z_{k+1}^{(S)} \to \cdots \to z_{k+1}^{(0)} \equiv x_{k+1}$. Our controller u_ψ injects small affine residuals U_{k+1} into the latents (per sub-step), conditioned on the current state x_k and a preview buffer ω_k of upcoming observations in $\mathcal{W}_k$. This yields stable, feed-forward autoregressive assimilation without test-time optimization.

2 Method

Notation. Physical time indices $k \in \mathbb{N}$; DDIM denoising sub-steps $s \in \{S-1, \dots, 0\}$. We define the state space $\mathcal{X} \triangleq \mathbb{R}^{C\times H\times W}$, with states $x_k \in \mathcal{X}$. Observation arrival indices $\mathcal{T} \subseteq \mathbb{N}$ with observations $y_\mathcal{T} \triangleq \{y_t\}_{t\in\mathcal{T}}$. For measures P, Q on trajectory space, dP/dQ denotes the Radon–Nikodym derivative when $P \ll Q$. We use $\odot$ for Hadamard products; $\|\cdot\|_2$ is the Euclidean norm over all channels/pixels.

2.1 Problem: Chaotic Forecasting with Delayed, Sparse Observations

Open-loop forecasts of chaotic dynamics quickly drift from the ground truth. DA combats this by using observations to periodically steer the forecast back toward reality. In practice, observations are sparse and may be delayed relative to the simulator's internal step size, so the model can evolve for many steps without guidance. Purely retrospective corrections then arrive too late to prevent drift. Classical DA mitigates this by using windows of observations (e.g., fixed-lag smoothing). We adopt the same perspective via a *preview* regime: during each transition $x_k \to x_{k+1}$, the sampler can access a short lookahead window of upcoming observations, enabling small anticipatory corrections that improve consistency at the next arrivals while staying close to the unguided simulator.

We now formalize the ingredients of our approach.

2.1.1 Forecaster's Prior Dynamics

Given an initial distribution $p_0(x_0)$ and one-step transition kernels $q(x_{k+1} \mid x_k)$, the induced trajectory distribution over the first $n+1$ states is $Q(x_{0:n}) = p_0(x_0) \prod_{k=0}^{n-1} q(x_{k+1} \mid x_k), n \in \mathbb{N}$. In our setting, the kernel q is realized by a pretrained one-step diffusion forecaster (details in App. A). We denote this kernel as $q_\theta(x_{k+1} \mid x_k) \equiv \texttt{DDIM}_S(g_\theta; x_k)$, yielding

$$Q_\theta(x_{0:n}) = p_0(x_0) \prod_{k=0}^{n-1} q_\theta(x_{k+1} \mid x_k), \qquad \forall n \in \mathbb{N}. \tag{1}$$

The family $\{Q_\theta(x_{0:n})\}_{n\in\mathbb{N}}$ is consistent and induces a semi-infinite path measure Q_θ^∞ on $\mathcal{X}^\mathbb{N}$, corresponding to an infinite-horizon autoregressive process.

2.1.2 Preview Selector: Windowed, Multi-Observation

Observations arrive only at sparse indices $\mathcal{T} \subseteq \mathbb{N}$, so forecasts can drift between arrivals. We therefore allow a bounded *preview*: at each transition $x_k \to x_{k+1}$, the controller may condition on upcoming observations within a fixed horizon Λ, analogous to finite assimilation windows (e.g., fixed-lag smoothing in 4D-Var and conditional lookahead in Shysheya et al. [10]).

Anchored preview windows. Let k_0 be the rollout start and generate $X_{k_0+1:k_0+\Lambda}$. For step $k \in \{k_0, \dots, k_0 + \Lambda - 1\}$, define

$$\mathcal{W}_{k|k_0} \triangleq \{\, j \in \mathcal{T} : k{+}1 \le j \le k_0{+}\Lambda \,\}, \qquad \Delta_{k,j} \triangleq j - (k{+}1), \tag{2}$$

and preview context $\omega_{k|k_0} \triangleq \{(y_j, \Delta_{k,j}) : j \in \mathcal{W}_{k|k_0}\}$. In experiments we use the nearest-arrival variant $\omega^\star_{k|k_0} = \{(y_{j^\star}, \Delta_{k,j^\star})\}$ with $j^\star = \min \mathcal{W}_{k|k_0}$ when nonempty. Online filtering is recovered with $\Lambda = 0$. Observation-operator metadata (e.g., masks M_j) accompanies y_j for use in Φ_j (App. B, App. C).

2.1.3 Observations as Arrival-Time Costs

Observations arrive at sparse indices $\mathcal{T} \subseteq \mathbb{N}$. At each arrival $k \in \mathcal{T}$ we impose an *arrival-time cost*

$$\mathcal{C}(x) \triangleq \sum_{k \in \mathcal{T}} \Phi_k(x_k;\, y_k), \tag{3}$$

where Φ_k penalizes mismatch between forecast x_k and observation y_k. Our objective trades off minimizing $\mathcal{C}(x)$ with staying close to the unguided simulator (Sec. 2.2). In our experiments, Φ_k uses linear masking/downsampling with least-squares penalties (App. B); more generally we only require $\Phi_k(\cdot; y_k)$ to be differentiable in x_k to train the controller.

Takeaway. Our setting combines (i) a diffusion path measure Q^∞_θ, (ii) a finite-horizon preview selector, and (iii) arrival-time costs tying forecasts to sparse observations. We next derive their optimal combination via exponential tilting.

2.2 Variational Principle: Gibbs Tilt for DA

We bias diffusion trajectories toward observations by reweighting the baseline path measure Q^∞_θ (Eq. (1)) using arrival-time costs (Eq. (3)). This yields the exponentially tilted (Gibbs) posterior

$$\frac{dP^\star_\beta}{dQ^\infty_\theta}(x) = \frac{e^{-\beta \mathcal{C}(x)}}{Z_\beta}, \qquad Z_\beta = \mathbb{E}_{Q^\infty_\theta}\left[e^{-\beta \mathcal{C}}\right], \tag{4}$$

assuming $0 < Z_\beta < \infty$ (e.g., under mild integrability of Φ_k and sparse/summable arrivals). The family $\{P^\star_\beta\}_{\beta>0}$ is the Gibbs posterior with temperature β^{-1}.

Gibbs variational characterization. The Gibbs variational principle gives

$$\log Z_\beta = \sup_{P \ll Q^\infty_\theta} \left\{ -\beta\, \mathbb{E}_P[\mathcal{C}] - \mathrm{KL}(P \| Q^\infty_\theta) \right\}, \tag{5}$$

with equality at $P = P^\star_\beta$, implying for any $P \ll Q^\infty_\theta$ the bound

$$-\beta\, \mathbb{E}_P[\mathcal{C}] - \mathrm{KL}(P \| Q^\infty_\theta) \le \log Z_\beta, \tag{6}$$

i.e., approaching $P^\star_\beta$ requires simultaneously low expected cost and small deviation from the baseline (proof in App. D).

Why we need a controlled variational family. Although $P^\star_\beta \propto e^{-\beta \mathcal{C}(x)} Q^\infty_\theta$ is optimal, exact sampling or estimating Z_β is intractable for autoregressive DDIM: $\mathcal{C}(x)$ requires full rollouts, and in chaotic regimes importance weights collapse. We therefore introduce a tractable family P_ψ by injecting preview-aware residual controls into Q^∞_θ (Sec. 2.3), preserving the diffusion backbone while reducing arrival-time costs without costly Gibbs sampling.

2.3 Approximation: Amortized Preview-Aware Control Family P_ψ

Motivation for Amortization. One could optimize controls per trajectory at test time, e.g., run a few inner iterations at each denoising step with future controls set to zero. This mirrors NDTM [12] in non-autoregressive images, where arrival costs can be estimated from intermediate noisy states (via Tweedie corrections) without full rollouts. In autoregressive forecasting this is not possible: x_k at any arrival $k \in \mathcal{T}$ depends on the entire preceding trajectory, so evaluating $\Phi_k(x_k; y_k)$ requires explicit rollouts through intermediate steps. Even a few inner iterations would therefore require repeated full rollouts, making test-time optimization infeasible. We instead *amortize* control selection by training a lightweight policy u_ψ offline on short preview rollouts, enabling single-pass, feed-forward control at test time while keeping the diffusion backbone frozen for stability and expressivity.

Controlled path measure. We retain the baseline sampler Q_θ and perturb only the *parent input* of each denoising sub-step through a small preview-aware map f_s. Formally, for latent variables $z_{k+1}^{(S)}, \dots, z_{k+1}^{(0)}$ with $z_{k+1}^{(0)} \equiv x_{k+1}$, the baseline one-step kernel factors as

$$q_\theta(x_{k+1} \mid x_k) \;=\; \int \left[\prod_{s=S-1}^{0} q_\theta^{(s)}\big(z_{k+1}^{(s)} \mid z_{k+1}^{(s+1)};\, x_k\big) \right] p_S\big(z_{k+1}^{(S)}\big)\; dz_{k+1}^{(1:S)}, \tag{7}$$

with p_S denoting the noise prior. Given the active preview ω_k (Sec. 2.1.2, App. C), the policy u_ψ (more details in App. F) emits control vectors

$$U_{k+1} = (u_{k+1}^{(S-1)}, \dots, u_{k+1}^{(0)}), \qquad u_{k+1}^{(s)} = u_\psi\big(x_k,\, \omega_{k|k_0},\, s\big),$$

which enter through the affine perturbation

$$f(z, u) \;=\; z + \gamma\, u, \qquad \gamma > 0. \tag{8}$$

Each controlled sub-step is then defined as

$$p_\psi^{(s)}\big(z_{k+1}^{(s)} \mid z_{k+1}^{(s+1)};\, u_{k+1}^{(s)}, x_k\big) \;\triangleq\; q_\theta^{(s)}\Big(z_{k+1}^{(s)} \,\Big|\, f(z_{k+1}^{(s+1)}, u_{k+1}^{(s)});\; x_k\Big). \tag{9}$$

Composing across s yields the controlled one-step kernel

$$p_\psi(x_{k+1} \mid x_k;\, U_{k+1}) \;=\; \int \left[\prod_{s=S-1}^{0} p_\psi^{(s)}\big(z_{k+1}^{(s)} \mid z_{k+1}^{(s+1)};\, u_{k+1}^{(s)}, x_k\big) \right] p_S\big(z_{k+1}^{(S)}\big)\; dz_{k+1}^{(1:S)}. \tag{10}$$

By Kolmogorov consistency, the controlled kernels define the semi-infinite process

$$P_\psi^\infty(x_{0:\infty} \mid y_{\mathcal{T}}) \;=\; p_0(x_0) \prod_{k \ge 0} p_\psi\big(x_{k+1} \,\big|\, x_k;\, U_{k+1}\big). \tag{11}$$

From principle to a learnable objective. Restricting (5) to the controlled family yields

$$\min_\psi \; \beta\, \mathbb{E}_{P_\psi^\infty}\big[\mathcal{C}(x)\big] \;+\; \mathrm{KL}\big(P_\psi^\infty \,\big\|\, Q_\theta^\infty\big)\,, \tag{12}$$

trading observation fidelity against deviation from the pretrained diffusion. Since the pathwise KL is intractable for autoregressive DDIM, we optimize a finite-window surrogate consistent with the preview protocol.

Windowed surrogate. For rollout start k_0 and horizon Λ, define arrivals $\mathcal{A}_{k_0,\Lambda} \triangleq \mathcal{T} \cap [k_0{+}1,\, k_0{+}\Lambda]$ and windowed cost $\mathcal{C}_{[k_0,\Lambda]}(x) \triangleq \sum_{j \in \mathcal{A}_{k_0,\Lambda}} \Phi_j(x_j; y_j)$. We minimize

$$\min_\psi \; \mathbb{E}\left[\frac{1}{\max\{|\mathcal{A}_{k_0,\Lambda}|, 1\}}\, \mathcal{C}_{[k_0,\Lambda]}(X^\psi) \;+\; \frac{1}{\beta}\, \mathcal{R}_{\mathrm{div}}(\psi; k_0, \Lambda) \right], \tag{13}$$

where $X^\psi = X^\psi_{k_0+1:k_0+\Lambda}$ is generated under the controlled kernel p_ψ using the anchored preview $\omega_{k|k_0}$. The regularizer $\mathcal{R}_{\mathrm{div}}$ is a tractable proxy for divergence from Q_θ (e.g., control energy $\|U\|_2^2$ or per-step proximity with shared noise seeds [12]), serving as the finite-window analogue of (12).

2.4 Algorithm: Preview-Aware Sampler & Trainer

Training: Windowed Rollouts with Arrival Supervision. We sample windows of length Λ, roll out with preview-aware injections, and minimize (14) in Alg. 1 (App. E). This aligns the training loss with the evaluation protocol (arrival-only supervision) and makes the learned controller causal with respect to the preview window.

Inference: Preview-Aware Sampler. At test time we run the preview-aware selector once per physical step and advance the sampler in Alg. 2, injecting residuals via (8). When generating L frames with $L > \Lambda$, we generate in *moving-window* chunks of length Λ, carrying the final state of chunk j as the initial state of chunk $j{+}1$; this mirrors operational overlapping-window DA [13].

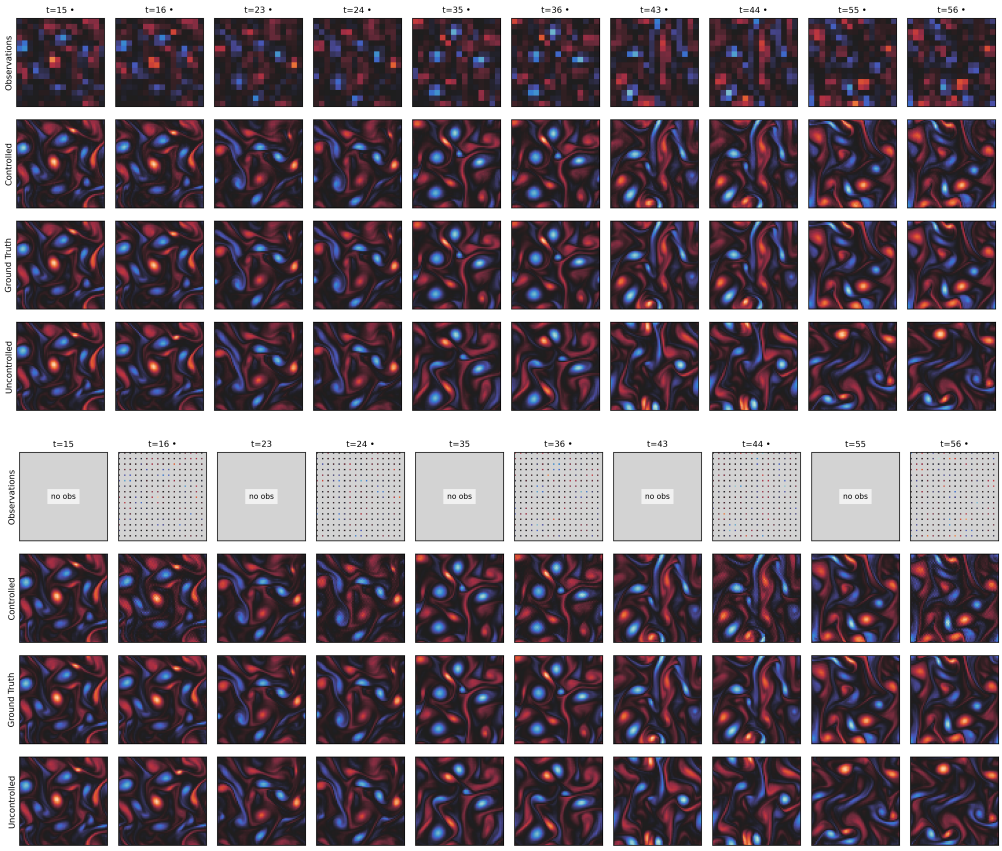

Figure 2: **Qualitative DA on Kolmogorov flow.** Rows: *Observations*, *Controlled* (ours), *Ground Truth*, *Uncontrolled*. Bullets mark observation arrivals. Uncontrolled forecasts are plausible until $t \approx 35$–36, then chaotic phase errors amplify, causing large structural drift for $t \geq 40$. Our preview-aware controller injects small corrections to stabilize rollouts, preserving phase coherence and fine-scale vortices. Top: Downsample $\times 4$ (denser). Bottom: Masked stride 4 (sparser).

3 Experiments

3.1 Kolmogorov Flow

We evaluate on 2D Kolmogorov flow, a standard turbulent PDE benchmark. The incompressible Navier–Stokes dynamics on $[0, 2\pi]^2$ with periodic boundaries are $\dot{u} = -u\nabla u + \frac{1}{\mathrm{Re}}\nabla^2 u - \frac{1}{\rho}\nabla p + f, \quad \nabla \cdot u = 0$, with $\mathrm{Re}{=}10^3$, $\rho{=}1$, and Kolmogorov forcing with linear damping. We generate trajectories with `jax-cfd` on a $256{\times}256$ grid, coarsened to $64{\times}64$, with snapshots every $\Delta{=}0.2$ (82 forward–Euler substeps). We simulate 1024 independent length-64 trajectories from the statistically stationary regime and split $80/10/10$ into train/val/test. Each state is a two-channel (u_x, u_y) field.

3.2 Observation Scenarios and Preview

Training uses a preview horizon $\Lambda{=}17$ (index 0 seeds autoregression). We study four observation operators, which define both training and evaluation tasks: (i) **Downsample** $\times 2$ and **Downsample** $\times 4$: observations at all preview indices $1{:}16$ and (ii) **Masked (stride 2)** and **Masked (stride 4)**: observations only at indices $\{4, 8, 12, 16\}$.

Across all experiments we instantiate the preview policy with the *nearest upcoming observation*. At physical step k, with lookahead horizon Λ, we select

$$k^\star \;=\; \arg\min_{j \in \mathcal{T} \cap \{k+1,\ldots,k+\Lambda\}} \big(j - (k{+}1)\big), \qquad \omega_k \;=\; \begin{cases} (y_{k^\star},\, \Delta_{k,k^\star}) & \text{if such } k^\star \text{ exists,} \\ \varnothing & \text{otherwise,} \end{cases}$$

Table 1: RMSD (lower is better) across observation tasks.

Method	Masked (s=2)	Masked (s=4)	Downsample ($\times 2$)	Downsample ($\times 4$)
SDA (AAO)	0.1411	0.3529	0.0413	0.2099
Joint AR	0.0429	0.1495	0.0383	0.1846
Ours	**0.0151**	**0.0223**	**0.0152**	**0.0220**

Table 2: Sampling time (seconds)

Method	Masked Obs.	Downsampled Obs.
SDA (AAO)	244.10	248.24
Joint AR	119.80	120.43
Ours	**3.55**	**3.86**

where $\Delta_{k,k^\star} = k^\star - (k{+}1)$ is the lead time. When $\omega_k = \varnothing$ the controller receives no preview and the step reduces to the frozen backbone transition. This nearest-arrival policy yields a constant-time, causal selector per step and keeps the controller lightweight. Extending to multi-arrival aggregation is supported by the formulation (Sec. 2.1.2) but is not used in our reported results.

3.3 Training Objective

In our implementation we operate in the high-β regime, yielding

$$\min_{\psi} \mathbb{E}\left[\frac{1}{\max\{|\mathcal{T} \cap \mathcal{W}_{k_0}|, 1\}} \mathcal{C}_{[k_0,\Lambda]}(X^\psi)\right], \qquad X^\psi \sim P_\psi^\infty(\cdot \mid y_\mathcal{T}; k_0, \Lambda). \tag{14}$$

We rely on a *small-gain design*—small control scale γ, a limited number of DDIM sub-steps ($S{=}3$ per physical step in all experiments), and near-zero control initialization—to keep P_ψ^∞ close to Q_θ^∞ in practice while still achieving substantial cost reduction.[1] Empirically this provides stable long-horizon rollouts without an explicit divergence term, while preserving the principled variational view through (12)–(13).

3.4 Metrics, Protocol, and Baselines

Protocol. We roll out $L{=}60$ steps autoregressively. Unless otherwise stated, all methods use identical observation streams and are evaluated with root mean square deviation (*RMSD*) over the forecast horizon; we report per-task means across the test set.

Baselines. (1) *SDA [7]*: score-based data assimilation that samples *all-at-once* trajectories and applies observation guidance at inference; this decouples observation models from training and enables zero-shot observation types but requires iterative, window-level denoising at test time. (2) *Joint AR [10]*: the "joint" score model conditioned on history but sampled *autoregressively* with reconstruction guidance; compared to AAO, AR improves forecasting stability while keeping the same score parameterization. These reflect current practice in diffusion-for-PDE DA and provide complementary trade-offs between conditioning flexibility and rollout stability.

3.5 Results

Accuracy. Table 1 reports RMSD across the four tasks. The preview-aware controller achieves the lowest error in all scenarios, with the largest margins under sparse masked observations.

Efficiency. We benchmark wall-clock sampling on a single RTX A6000 for 10 assimilated trajectories (data loading and metric computation excluded). Amortized preview control yields markedly lower runtime than both baselines (Table 2), translating to $30\times$–$70\times$ speedups depending on the baseline.

Takeaway. Across masking and downsampling tasks, amortized preview control combines the stability of a frozen diffusion backbone with lightweight, lookahead corrections. This yields consistent accuracy gains and large end-to-end speedups, making assimilation a single forward rollout rather than a test-time optimization or ensemble computation. Fig. 2 clearly states the efficacy of such controlled autoregressive diffusion models in preventing trajectory divergence.

[1] See App. G for implementation notes (gradient checkpointing across UNet calls) and App. F for the ControlNet/UNet specifications.

References

[1] Hao Wang, Jindong Han, Wei Fan, Weijia Zhang, and Hao Liu. Phyda: Physics-guided diffusion models for data assimilation in atmospheric systems. *arXiv preprint arXiv:2505.12882*, 2025.

[2] Alberto Carrassi, Marc Bocquet, Laurent Bertino, and Geir Evensen. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *WIREs Climate Change*, 9(5): e535, 2018. doi: https://doi.org/10.1002/wcc.535. URL `https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcc.535`.

[3] François-Xavier Le Dimet and Olivier Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, 38(2):97–110, 1986.

[4] Philippe Courtier, E Andersson, W Heckley, D Vasiljevic, M Hamrud, A Hollingsworth, F Rabier, M Fisher, and J Pailleux. The ecmwf implementation of three-dimensional variational assimilation (3d-var). i: Formulation. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1783–1807, 1998.

[5] 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.

[6] Rui Wang and Rose Yu. Physics-guided deep learning for dynamical systems: A survey. *arXiv preprint arXiv:2107.01272*, 2021.

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

[8] 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 (CVPR) Workshops*, pages 449–459, June 2024.

[9] Peter Manshausen, Yair Cohen, Peter Harrington, Jaideep Pathak, Mike Pritchard, Piyush Garg, Morteza Mardani, Karthik Kashinath, Simon Byrne, and Noah Brenowitz. Generative data assimilation of sparse weather station observations at kilometer scales, 2025. URL `https://arxiv.org/abs/2406.16947`.

[10] Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard Turner, and Emile Mathieu. On conditional diffusion models for pde simulations. *Advances in Neural Information Processing Systems*, 37:23246–23300, 2024.

[11] Langwen Huang, Lukas Gianinazzi, Yuejiang Yu, Peter D Dueben, and Torsten Hoefler. Diffda: a diffusion model for weather-scale data assimilation. *arXiv preprint arXiv:2401.05932*, 2024.

[12] Kushagra Pandey, Farrin Marouf Sofian, Felix Draxler, Theofanis Karaletsos, and Stephan Mandt. Variational control for guidance in diffusion models. *arXiv preprint arXiv:2502.03686*, 2025.

[13] Laura C Slivinski, Donald E Lippi, Jeffrey S Whitaker, Guoqing Ge, Jacob R Carley, Curtis R Alexander, and Gilbert P Compo. Overlapping windows in a global hourly data assimilation system. *Monthly Weather Review*, 150(6):1317–1334, 2022.

A DDIM parameterization, coefficients, SNR, and $v \to \varepsilon$

At denoising step s, DDIM yields a Gaussian transition

$$x^{(s-1)} \sim \mathcal{N}\big(\,\mu_\theta(x^{(s)}, s),\ \sigma_s^2 I\,\big), \qquad \mu_\theta(x^{(s)}, s) \;=\; a_s\, x^{(s)} + b_s\, \widehat{\varepsilon}_\theta(x^{(s)}, s),$$

with schedule-dependent (a_s, b_s, σ_s); deterministic DDIM uses $\sigma_s{=}0$. We pass $\log \mathrm{SNR}(s) = \log \frac{\bar{\alpha}_s}{1-\bar{\alpha}_s}$ to the control. Our UNet is trained with v-prediction; we convert to noise prediction via

$$\boxed{\widehat{\varepsilon}_\theta(x^{(s)}, s) = \sqrt{\bar{\alpha}_s}\, \widehat{v}_\theta(x^{(s)}, s) + \sqrt{1-\bar{\alpha}_s}\; x^{(s)}}$$

and use $\widehat{\varepsilon}_\theta$ in all DDIM formulas.

B Observation operators used in experiments

We instantiate terminal costs using simple linear observation operators for clarity and stability. For a (possibly time-varying) mask $M \in \{0,1\}^{1\times H\times W}$ broadcast across channels,

$$A_{\text{mask}}(x) = M \odot x, \quad \Phi_k^{\text{mask}}(x;y) = \frac{\|M \odot (x-y)\|_2^2}{\|M\|_1 + \varepsilon},$$

with $\varepsilon = 10^{-6}$ and the step skipped if $\|M\|_1 = 0$. For downsample/upsample we use average pooling P_f over non-overlapping $f \times f$ blocks and nearest-neighbor upsampling U_f:

$$A_{\downarrow f}(x) = U_f(P_f x), \quad \Phi_k^{\text{ds}}(x;y) = \big\|U_f(P_f x) - U_f(P_f y)\big\|_2^2.$$

These operators are used to generate the observed signals y_k that enter terminal costs. Any differentiable Φ_k could replace these without changing the method.

C Active observation selector (preview DA)

In practice, we maintain a preview buffer containing future observations from $\mathcal{T}$ that lie within the lookahead horizon Λ. Each entry is a triplet (y_j, M_j, Δ_j), where:

- $j \in \mathcal{T}$ is the physical time index of the observation,
- y_j is the observed signal (lifted to full resolution if needed),
- M_j is an auxiliary mask (binary for masking operators, all ones for downsampling; see App. B; for other operators, M_j may be ignored or replaced by auxiliary metadata as appropriate),
- $\Delta_j = j - k + 1$ is the lead time relative to the current forecast step k.

At each physical step k, the active preview is chosen by

$$k^\star \;=\; \arg\min_{j \in \mathcal{T}\cap\mathcal{W}_k} \{\Delta_j : \Delta_j \geq 0\},$$

where $\mathcal{W}_k = \{k+1, \ldots, k+\Lambda\}$ is the preview window. The selected preview is then

$$\omega_k = (y_{k^\star}, M_{k^\star}, \Delta_{k^\star}),$$

which is passed to the controller at step k. This selection occurs *once per physical step*.

D Gibbs variational principle (proof)

Let $(\mathcal{X}^{\mathbb{N}}, \mathcal{F})$ denote the trajectory space with its product σ-algebra, and let Q_θ^∞ be the baseline path measure from Eq. (1). Fix a measurable cost $\mathcal{C} : \mathcal{X}^{\mathbb{N}} \to \mathbb{R}$ and $\beta > 0$, and assume the mild integrability condition

$$0 \;<\; Z_\beta \;\triangleq\; \mathbb{E}_{Q_\theta^\infty}\big[e^{-\beta\mathcal{C}}\big] \;<\; \infty.$$

Define the exponentially tilted (Gibbs) measure $P_\beta^\star$ by

$$\frac{dP_\beta^\star}{dQ_\theta^\infty}(x) \;=\; \frac{e^{-\beta\mathcal{C}(x)}}{Z_\beta}. \tag{15}$$

Variational identity. For any $P \ll Q_\theta^\infty$,

$$\begin{aligned}
\mathrm{KL}\big(P\big\|P_\beta^\star\big) &= \int \log\left(\frac{dP}{dP_\beta^\star}\right) dP \;=\; \int \log\left(\frac{dP/dQ_\theta^\infty}{dP_\beta^\star/dQ_\theta^\infty}\right) dP \\
&= \int \log\Big(\tfrac{dP}{dQ_\theta^\infty} \cdot Z_\beta\, e^{\beta\mathcal{C}}\Big)\, dP \\
&= \mathrm{KL}(P\|Q_\theta^\infty) \;+\; \beta\,\mathbb{E}_P[\mathcal{C}] \;+\; \log Z_\beta.
\end{aligned}$$

Since $\mathrm{KL}(P\|P_\beta^\star) \geq 0$, we obtain

$$-\beta\,\mathbb{E}_P[\mathcal{C}] \;-\; \mathrm{KL}(P\|Q_\theta^\infty) \;\leq\; \log Z_\beta, \qquad \forall\, P \ll Q_\theta^\infty, \tag{16}$$

which is Eq. (6) in the main text.

Algorithm 1 Preview-Aware Control Training (Windowed Arrival Supervision)

Require: Frozen pretrained diffusion forecaster g_θ with S DDIM sub-steps; control policy network u_ψ with parameters ψ to be trained; preview horizon length Λ; control scale factor γ; arrival indices $\mathcal{T}$ with costs $\{\Phi_k\}_{k\in\mathcal{T}}$; initial distribution p_0 or dataset states.

Initialize ψ.
while not converged **do**
  Sample a start index k_0 and an initial state x_{k_0}.
  Define the training window $\{k_0{+}1, \dots, k_0{+}\Lambda\}$ and preview sets $\mathcal{W}_k = \{\, j \in \mathcal{T} \,:\, 1 \le j-(k{+}1) \le \Lambda \,\}$ for $k \in \{k_0, \dots, k_0{+}\Lambda{-}1\}$ (Sec. 2.1.2).
  $x \leftarrow x_{k_0}$, $\mathcal{L} \leftarrow 0$.
  for $k = k_0, \dots, k_0{+}\Lambda{-}1$ **do**
    Form the active preview $\omega_k = \{(y_j, \Delta_{k,j}) : j \in \mathcal{W}_k\}$ with $\Delta_{k,j} = j-(k{+}1)$.
    Compute controls $U_{k+1} = (u_{k+1}^{(S-1)}, \dots, u_{k+1}^{(0)})$ with $u_{k+1}^{(s)} = u_\psi(x, \omega_k, s)$.
    Generate x_{k+1} by composing controlled sub-steps (9) with map $f(z,u) = z + \gamma u$ (8):

$$z_{k+1}^{(S)} \sim p_S, \quad z_{k+1}^{(s)} \sim q_\theta^{(s)}\big(\cdot \mid f(z_{k+1}^{(s+1)}, u_{k+1}^{(s)});\, x\big),\ s{=}S{-}1{:}0, \quad x_{k+1} \equiv z_{k+1}^{(0)}.$$

    $x \leftarrow x_{k+1}$.
    if $k{+}1 \in \mathcal{T}$ **then**
      $\mathcal{L} \leftarrow \mathcal{L} + \Phi_{k+1}(x_{k+1}; y_{k+1})$ ▷ arrival-only supervision (3)
  Normalize window loss $\tilde{\mathcal{L}} = \dfrac{\mathcal{L}}{\max\{|\mathcal{T} \cap \mathcal{W}_{k_0}|,\, 1\}}$ and update $\psi \leftarrow \psi - \eta_\psi \nabla_\psi \tilde{\mathcal{L}}$ ▷ g_θ frozen; backprop through controlled DDIM
return Trained control parameters $\psi^\star$.

Algorithm 2 Preview-Aware Amortized Assimilation (Inference)

Require: Frozen g_θ; trained controller $u_{\psi^\star}$; control scale γ; preview horizon Λ; initial state x_0; observation stream $\{(y_k, \Phi_k)\}_{k\in\mathcal{T}}$.

for $k = 0, 1, \dots, L{-}1$ **do**
  Build preview set $\mathcal{W}_k = \{\, j \in \mathcal{T} \,:\, 1 \le j-(k{+}1) \le \Lambda \,\}$ and active preview $\omega_k = \{(y_j, \Delta_{k,j})\}_{j\in\mathcal{W}_k}$.
  Compute controls $U_{k+1} = (u_{k+1}^{(S-1)}, \dots, u_{k+1}^{(0)})$ with $u_{k+1}^{(s)} = u_{\psi^\star}(x_k, \omega_k, s)$.
  Advance one physical step using the controlled kernel (10):

$$z_{k+1}^{(S)} \sim p_S, \quad z_{k+1}^{(s)} \sim q_\theta^{(s)}\big(\cdot \mid f(z_{k+1}^{(s+1)}, u_{k+1}^{(s)});\, x_k\big),\ s{=}S{-}1{:}0, \quad x_{k+1} \equiv z_{k+1}^{(0)}.$$

  Optionally discard arrivals $j \le k{+}1$ from the stream; set conditioner $x_k \leftarrow x_{k+1}$.
return Forecast path $x_{1:L}$.

Optimality and uniqueness. Equality in (16) holds iff $\mathrm{KL}(P\|P_\beta^\star) = 0$, i.e., iff $P = P_\beta^\star$ (equality Q_θ^∞-a.s.). Equivalently,

$$\log Z_\beta \;=\; \sup_{P \ll Q_\theta^\infty} \Big\{ -\beta\, \mathbb{E}_P[\mathcal{C}] \;-\; \mathrm{KL}(P\|Q_\theta^\infty) \Big\}, \tag{17}$$

and the unique maximizer is $P_\beta^\star$.

Remarks. (i) The same proof applies verbatim on any finite horizon by replacing Q_θ^∞ and $\mathcal{C}$ with their restrictions to $\mathcal{X}^{0:n}$, yielding the identical identity and optimizer. (ii) If $P \not\ll Q_\theta^\infty$, interpret $\mathrm{KL}(P\|Q_\theta^\infty) = +\infty$, so such P do not affect the supremum in (17).

E Algorithm

Alg. 1 and Alg. 2 are the training and inference algorithm respectively. While the model has been trained on a small preview window, we test the performance on rollouts far larger than that.

F Control network implementation

Purpose. The control policy u_ψ generates the residual control $u_k^{(s)}$. In Sec. 2.3, we write $u_{k+1}^{(s)} = u_\psi(x_k, \omega_k, s)$ for clarity. Here we expand ω_k and the additional inputs required in practice. Formally,

$$u_k^{(s)} \;=\; u_\psi\Big(x_{k+1}^{(s)},\; x_k,\; y_k^\star,\; M_k^\star,\; \Delta_k^\star,\; \log \mathrm{SNR}(s),\; \tau,\; u_{\mathrm{prev}}\Big).$$

Inputs and fusion. We concatenate five image-like tensors along channels: the current latent $x_{k+1}^{(s)}$, the previous state x_k, the preview observation $y_k^\star$, the auxiliary mask $M_k^\star$, and the previous control u_{prev}. A shallow encoder with a two-level down/up path extracts limited spatial context. FiLM modulation injects scalar metadata $(\Delta_k^\star, \tau, \log \mathrm{SNR}(s))$ where $\Delta_k^\star$ is the preview lag and τ is the local position index in the Λ window $\mathcal{W}_k$.

FiLM conditioning. Each scalar is normalized and embedded by an MLP: $\Delta_k^\star/\Lambda$, τ/Λ, and $\log \mathrm{SNR}(s)$. The embeddings are concatenated and mapped to (γ, β), which modulate feature maps as feat $\mapsto$ feat $\cdot\, (1{+}\gamma) + \beta$.

Residual head and stability. A $3{\times}3$ convolutional head outputs Δ_ψ, which is added to a normalized copy of u_{prev} to yield $u_k^{(s)}$. At the first denoising sub-step ($s{=}S{-}1$), we set $u_{\mathrm{prev}} = 0$.

Usage notes. We normalize $(\Delta_k^\star, \tau)$ to $[0, 1]$, and compute $\log \mathrm{SNR}(s)$ from the current DDIM schedule (App. A). This design keeps u_ψ lightweight relative to the UNet backbone while expressive enough to bias forecasts toward observations.

G Implementation notes

Gradients flow only into ψ (the UNet θ is frozen). We use gradient checkpointing at each UNet call and detach u_{prev} within a frame to avoid deep denoising-step recurrences; memory scales with the number of checkpoints.