Control-Augmented Diffusion for Autoregressive Data Assimilation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data assimilation (DA) in chaotic spatiotemporal systems, such as turbulent 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 observation gaps. At inference, assimilation reduces to a single forward rollout with on-the-fly corrections, avoiding optimization 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 spatiotemporal dynamics, from turbulence to weather, is notoriously difficult due to chaos: small state errors amplify exponentially, causing open-loop forecasts to diverge. Data assimilation (DA) counters this by incorporating sparse, noisy observations, producing improved analyses that extend predictability [1, 2]. Classical schemes such as 4D-Var and EnKF [3, 4, 5] have long powered operational forecasting, but rely on quasi-linear assumptions and demand costly adjoint or ensemble computations [6].

Deep generative models offer an alternative. Diffusion models in particular capture high-dimensional distributions and can reconstruct full states from partial data. Recent works apply diffusion for Bayesian DA, either by guiding sampling at test time [7, 8, 9] or by conditioning training directly on observations [10, 11]. While promising, these approaches remain limited: guidance applied only at inference allows errors to accumulate between arrivals; naïve conditional training destabilizes long rollouts; and iterative denoising makes inference slow.

We propose a diffusion-based DA framework that introduces a learned control mechanism into the generative dynamics. A pretrained diffusion forecaster provides the backbone transitions, while a control network injects affine residuals into each DDIM step (Fig. 1). These residuals act as lightweight, preview-aware corrections, nudging the trajectory toward upcoming observations without altering the backbone. Crucially, the controller is trained offline on synthetic assimilation scenarios, so at test time the system performs causal, feed-forward rollouts with on-the-fly corrections. This amortized design combines the expressivity of diffusion models with the efficiency of learned control, enabling accurate, stable, and fast assimilation in autoregressive chaotic PDE forecasting.

Submitted to the AI for Science workshop (NeurIPS 2025).

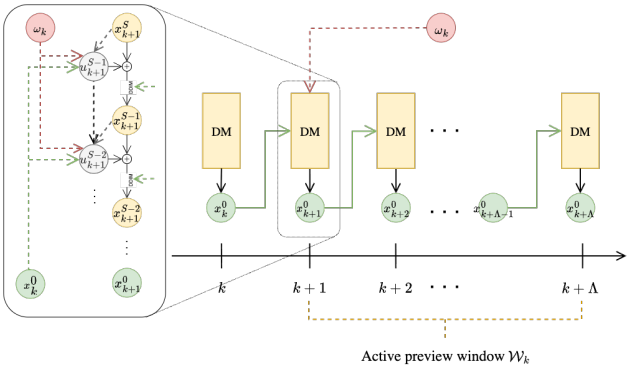

Figure 1: At physical step k, the pretrained diffusion backbone (DM) advances the state x_k through S denoising sub-steps to produce x_{k+1}. Internally, the latent chain $z_{k+1}^{(S)} \to \cdots \to z_{k+1}^{(0)} \equiv x_{k+1}$ is generated by the pretrained DDIM sampler. Our control policy u_ψ injects small residuals U_{k+1} (added affinely to the parent latent at sub-steps s) that gently nudge the denoiser towards corrected trajectory. These residuals depend on the current state x_k and a preview buffer ω_k, which collects upcoming observations within the lookahead horizon $\mathcal{W}_k$. This design enables stable assimilation through feed-forward autoregressive rollouts 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 neural simulators of chaotic dynamics inevitably diverge exponentially from ground-truth trajectories. Data assimilation (DA) mitigates this instability by periodically steering forecasts back toward reality using observational data or other guidance signals.

In practice, observations arrive far less frequently than the simulator's internal time step, and often with reporting delays. This creates a mismatch: the simulator advances many steps without guidance, so purely retrospective corrections cannot prevent substantial drift. Classical data assimilation addresses this by optimizing over windows of past and future observations (e.g., fixed-lag smoothing). Inspired by this view, we introduce a *preview* regime: during each transition $x_k \to x_{k+1}$, the sampler can access a short lookahead window of upcoming observations. These future cues allow the controller to apply small anticipatory corrections, nudging the forecast toward consistency at the next arrival while staying close to the unguided simulator.

With this intuition in place, we now introduce the formal ingredients of our approach.

2.1.1 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 in our setting occur only at a sparse subset of simulator steps $\mathcal{T} \subseteq \mathbb{N}$, leaving long stretches without direct guidance. Relying solely on past arrivals can therefore allow the simulator to drift substantially before the next observation is available. To address this, we introduce a bounded preview: during the transition $x_k \to x_{k+1}$, the controller is permitted to condition on nearby upcoming arrivals within a fixed horizon Λ. This design is analogous to finite assimilation windows in data assimilation (e.g., fixed-lag smoothing in 4D-Var/EnKS and conditional lookahead in 10).

Formally, for each step k the preview window is

$$\mathcal{W}_k \triangleq \{\, j \in \mathcal{T} : 1 \le j - (k{+}1) \le \Lambda \,\},$$

the set of arrival indices within the next Λ simulator steps. If $\mathcal{W}_k \neq \varnothing$, the *active preview* is

$$\omega_k \triangleq \{(y_j,\, \Delta_{k,j}) : j \in \mathcal{W}_k,\ \Delta_{k,j} = j - (k{+}1)\}.$$

Operator-specific metadata (e.g., masks M_j for masked losses) travel with y_j as needed to evaluate Φ_j, but are not part of the assimilation logic itself (see App. B, App. C). In experiments we restrict to the *nearest-observation* special case for efficiency, though the formulation supports aggregation over all elements of ω_k.

2.1.3 Observations as Arrival-Time Costs

Observations arrive at a sparse subset of simulator steps, indexed by $\mathcal{T} \subseteq \mathbb{N}$. At these arrival indices $k \in \mathcal{T}$ we impose *arrival-time costs*:

$$\mathcal{C}(x) \triangleq \sum_{k \in \mathcal{T}} \Phi_k(x_k;\, y_k). \tag{2}$$

Each Φ_k penalizes mismatch between forecast state x_k and its corresponding observation y_k, and our learning objective balances this cumulative arrival-time cost against remaining close to the unguided simulator (Sec. 2.2). Concrete operators used in our experiments are defined in App. B; while these are linear masking/downsampling operators instantiated with least-squares penalties, the only essential requirement for the framework is that $\Phi_k(\cdot\,; y_k)$ be differentiable in x_k so that gradients can train the controller.

Takeaway. Together, these ingredients define our assimilation setting: (i) a diffusion-based path measure Q_θ^∞ that generates forecasts over an infinite horizon, (iii) a preview selector that associates each simulator step with the nearest upcoming arrivals inside a finite lookahead window, and (ii) an arrival-time cost interface that ties those forecasts to sparse observations. The next step is to ask how to optimally combine these ingredients, which leads to a variational view via exponential tilting.

2.2 Variational Principle: Gibbs Tilt for DA

Given the baseline path measure Q_θ^∞ (Eq. (1)) and arrival-time costs $\mathcal{C}(x) \triangleq \sum_{k \in \mathcal{T}} \Phi_k(x_k; y_k)$, a natural way to bias trajectories toward observations is through an exponential tilt:

$$\frac{dP_\beta^\star}{dQ_\theta^\infty}(x) \;=\; \frac{\exp(-\beta \mathcal{C}(x))}{Z_\beta}, \qquad Z_\beta \;=\; \mathbb{E}_{Q_\theta^\infty}[\exp(-\beta \mathcal{C})]\,. \tag{3}$$

For this to define a valid probability measure we require $0 < Z_\beta < \infty$, which holds under mild integrability assumptions, e.g. bounded Φ_k or suitably sparse/summable observation schedules. Under these conditions, $\{P_\beta^\star\}_{\beta > 0}$ forms the Gibbs posterior family, with temperature β^{-1}.

Gibbs Variational Characterization. By the Gibbs variational principle,

$$\log Z_\beta \;=\; \sup_{P \ll Q_\theta^\infty} \big\{ -\beta\, \mathbb{E}_P[\mathcal{C}] \;-\; \mathrm{KL}(P \| Q_\theta^\infty) \big\}, \tag{4}$$

with equality attained at $P = P_\beta^\star$. Thus, for any $P \ll Q_\theta^\infty$ we obtain the bound

$$-\beta\, \mathbb{E}_P[\mathcal{C}] \;-\; \mathrm{KL}(P \| Q_\theta^\infty) \;\le\; \log Z_\beta. \tag{5}$$

This shows that small expected cost and small divergence from the baseline are jointly necessary: a distribution P can only approach the optimal Gibbs posterior by balancing both terms. A short proof of (4) and (5) is given in App. D.

Why Direct Tilt Is Intractable for Autoregressive DDIM. The Gibbs posterior $P_\beta^\star \propto e^{-\beta \mathcal{C}(x)} Q_\theta^\infty$ is in principle the optimal distribution, but computing its normalization constant Z_β or sampling from it exactly is intractable. A natural idea is importance sampling from Q_θ^∞, but in the autoregressive DDIM setting this quickly breaks down: evaluating the cost $\mathcal{C}(x)$ requires full autoregressive rollouts of the diffusion backbone, and in chaotic regimes the importance weights concentrate on a vanishing fraction of trajectories. Thus an astronomical number of rollouts would be needed to obtain reliable estimates. We therefore turn to variational inference: rather than reweighting, we define a tractable parametric family P_ψ by injecting preview-aware residual controls into Q_θ^∞ (Sec. 2.3). This retains the diffusion backbone for stability while enabling lightweight corrections that reduce arrival-time costs without incurring the prohibitive expense of direct Gibbs sampling.

2.3 Approximation: Amortized Preview-Aware Control Family P_ψ

Motivation for Amortization. A natural strategy within the control family is to optimize controls per trajectory at test time. At each denoising step the control vector can be initialized at zero, future controls assumed zero, and a few inner iterations carried out to improve the current control before proceeding. This is analogous to NDTM [12] in the non-autoregressive image setting, where the cost at an arrival index can be estimated directly from a noisy intermediate state via a Tweedie correction, thereby avoiding a full rollout. In the autoregressive forecasting setting, however, such shortcuts are unavailable: the state at an arrival index $k \in \mathcal{T}$ depends on the entire preceding trajectory, so evaluating $\Phi_k(x_k; y_k)$ requires an explicit rollout through all intermediate denoising steps. As a result, even a handful of inner optimization iterations per control would entail repeated full rollouts, which is computationally infeasible. We therefore *amortize* control selection: a lightweight policy u_ψ is trained offline on short preview rollouts so that, at test time, controls can be applied in a single forward pass per step. This design avoids costly trajectory-level optimization while retaining the frozen diffusion backbone 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

$$q_\theta(x_{k+1} \mid x_k) \;=\; \int \left[\prod_{s=S-1}^{0} q_\theta^{(s)}\big(z_{k+1}^{(s)} \mid z_{k+1}^{(s+1)};\, x_k\big) \right] p_S\big(z_{k+1}^{(S)}\big)\; dz_{k+1}^{(1:S)}, \tag{6}$$

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. E) emits control vectors

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

which enter through the affine perturbation

$$f(z, u) \;=\; z + \gamma\, u, \qquad \gamma > 0. \tag{7}$$

Each controlled sub-step is then defined as

$$p_\psi^{(s)}\big(z_{k+1}^{(s)} \mid z_{k+1}^{(s+1)};\, u_{k+1}^{(s)}, x_k\big) \;\triangleq\; q_\theta^{(s)}\Big(z_{k+1}^{(s)} \,\Big|\, f(z_{k+1}^{(s+1)}, u_{k+1}^{(s)});\; x_k\Big). \tag{8}$$

Composing across s yields the controlled one-step kernel

$$p_\psi(x_{k+1} \mid x_k;\, U_{k+1}) \;=\; \int \left[\prod_{s=S-1}^{0} p_\psi^{(s)}\big(z_{k+1}^{(s)} \mid z_{k+1}^{(s+1)};\, u_{k+1}^{(s)}, x_k\big) \right] p_S\big(z_{k+1}^{(S)}\big)\; dz_{k+1}^{(1:S)}. \tag{9}$$

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

$$P_\psi^\infty(x_{0:\infty} \mid y_\mathcal{T}) \;=\; p_0(x_0) \prod_{k \ge 0} p_\psi\big(x_{k+1} \,\big|\, x_k;\, U_{k+1}(x_k, \omega_k)\big). \tag{10}$$

From principle to a learnable objective. A direct instantiation of (4) with our family leads to the *principled* objective

$$\min_{\psi} \; \beta\, \mathbb{E}_{P_\psi^\infty}\big[\mathcal{C}(x)\big] \; + \; \mathrm{KL}\big(P_\psi^\infty \,\big\|\, Q_\theta^\infty\big)\,, \tag{11}$$

which trades off fidelity to observations against deviation from the frozen backbone. Computing the pathwise KL exactly is intractable for autoregressive DDIM; instead, we optimize a *windowed surrogate* aligned with preview rollouts. For a start index k_0 and horizon Λ, define

$$\mathcal{C}_{[k_0,\Lambda]}(x) \;=\; \sum_{k\in\mathcal{T}\cap\mathcal{W}_{k_0}} \Phi_k(x_k; y_k), \qquad \mathcal{W}_{k_0} = \{\, j \in \mathcal{T} : 1 \le j - (k_0{+}1) \le \Lambda \,\}.$$

We then minimize

$$\min_{\psi} \; \mathbb{E}\left[\underbrace{\frac{1}{\max\{|\mathcal{T}\cap\mathcal{W}_{k_0}|,1\}}\,\mathcal{C}_{[k_0,\Lambda]}(X^\psi)}_{\text{arrival-time cost over preview window}} \; + \; \underbrace{\frac{1}{\beta}\,\mathcal{R}_{\mathrm{div}}(\psi; k_0, \Lambda)}_{\text{divergence control (proxy)}} \right], \qquad X^\psi \sim P_\psi^\infty(\cdot \mid y_{\mathcal{T}}; k_0, \Lambda), \tag{12}$$

where $\mathcal{R}_{\mathrm{div}}$ is any tractable proxy that discourages large departures from Q_θ (e.g., control-energy $\|U\|_2^2$, or a per-step proximity penalty between controlled and baseline one-step predictions with a shared noise seed akin to Pandey et al. [12]). This surrogate is the finite-window counterpart of (11) and matches the causal preview protocol.

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 (13) in Alg. 1. 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 (7). 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].

3 Experiments

3.1 Kolmogorov Flow

We evaluate on the two–dimensional Kolmogorov flow, a standard turbulent PDE benchmark. Incompressible dynamics follow the Navier–Stokes equations on $[0, 2\pi]^2$ with periodic boundaries,

$$\dot{u} \;=\; -\,u\nabla u \;+\; \tfrac{1}{\mathrm{Re}}\nabla^2 u \;-\; \tfrac{1}{\rho}\nabla p \;+\; f, \qquad 0 \;=\; \nabla\cdot u,$$

with $\mathrm{Re}{=}10^3$, $\rho{=}1$, and Kolmogorov forcing with linear damping. We generate trajectories using `jax-cfd` on a $256{\times}256$ grid and coarsen states to $64{\times}64$. Snapshots are spaced by $\Delta = 0.2$ (82 forward–Euler substeps). We simulate 1024 independent length-64 trajectories from the statistically stationary regime and split into train/val/test as $80\%/10\%/10\%$. 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:

- **Downsample** $\times 2$ and **Downsample** $\times 4$: observations at all preview indices $1{:}16$.
- **Masked (stride 2)** and **Masked (stride 4)**: observations only at indices $\{4, 8, 12, 16\}$.

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

Require: Frozen diffusion forecaster g_θ with S DDIM sub-steps; control policy u_ψ with parameters ψ; preview horizon Λ; control scale 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 (8) with map $f(z,u) = z + \gamma u$ (7):

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

    $x \leftarrow x_{k+1}$.
    if $k{+}1 \in \mathcal{T}$ **then**
      $\mathcal{L} \leftarrow \mathcal{L} + \Phi_{k+1}(x_{k+1}; y_{k+1})$ ▷ arrival-only supervision (2)
  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 (9):

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

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

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

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

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

$$\min_\psi \; \mathbb{E}\left[\frac{1}{\max\{|\mathcal{T}\cap\mathcal{W}_{k_0}|, 1\}}\, \mathcal{C}_{[k_0,\Lambda]}(X^\psi)\right], \qquad X^\psi \sim P_\psi^\infty(\cdot \mid y_\mathcal{T}; k_0, \Lambda). \tag{13}$$

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

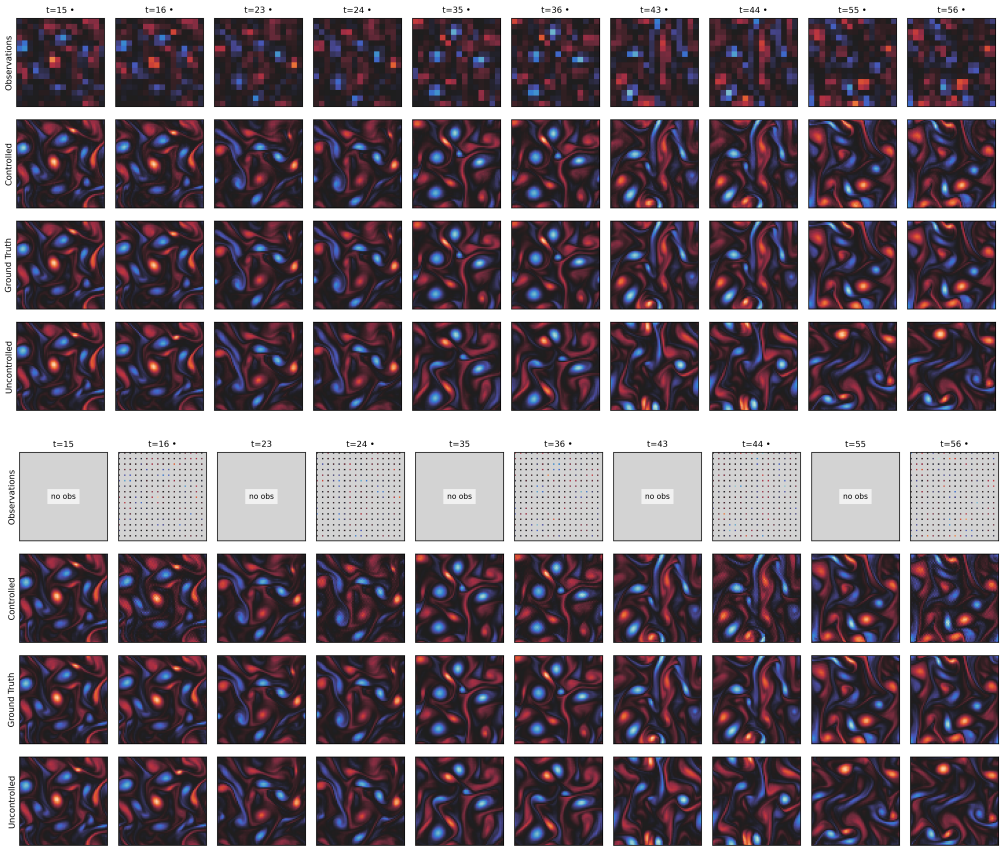

Figure 2: **Qualitative assimilation results on Kolmogorov flow.** Rows show *Observations*, *Controlled* (ours), *Ground Truth*, and *Uncontrolled* forecasts. Bullets above columns mark observation arrivals. Uncontrolled forecasts remain visually plausible up to about $t \approx 35$–36, after which chaotic divergence manifests: small phase errors amplify exponentially, leading to severe structural mismatches at later times ($t \geq 40$). Our preview-aware controller successfully stabilizes rollouts by injecting small corrections, maintaining phase coherence and preserving fine-scale vortical structures across the horizon. Panel (top): Downsample $\times 4$ (dense arrivals). Panel (bottom): Masked stride 4 (sparse arrivals).

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 (11)–(12).

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 *RMSD* (root mean square deviation) 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.

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

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) for 10 trajectories on one RTX A6000.

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

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.

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

$$\frac{dP_\beta^\star}{dQ_\theta^\infty}(x) \;=\; \frac{e^{-\beta \mathcal{C}(x)}}{Z_\beta}. \tag{14}$$

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

$$-\beta\, \mathbb{E}_P[\mathcal{C}] \;-\; \mathrm{KL}(P \| Q_\theta^\infty) \;\leq\; \log Z_\beta, \qquad \forall\, P \ll Q_\theta^\infty, \tag{15}$$

which is Eq. (5) in the main text.

Optimality and uniqueness. Equality in (15) holds iff $\mathrm{KL}(P \| P_\beta^\star) = 0$, i.e., iff $P = P_\beta^\star$ (equality Q_θ^∞-a.s.). Equivalently,

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

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 (16).

E 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.

F 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.