

Appa: Bending Weather Dynamics with Latent Diffusion Models for Global Data Assimilation

G r me Andry* Fran ois Rozet Sacha Lewin Omer Rochman Victor Mangeleer
Matthias Pirlet Elise Faulx Marilaure Gr goire Gilles Louppe

Abstract

Deep learning has transformed weather forecasting by improving both its accuracy and computational efficiency. However, before any forecast can begin, weather centers must identify the current atmospheric state from vast amounts of observational data. To address this challenging problem, we introduce Appa, a score-based data assimilation model producing global atmospheric trajectories at 0.25-degree resolution and 1-hour intervals. Powered by a 1.5B-parameter spatio-temporal latent diffusion model trained on ERA5 reanalysis data, Appa can be conditioned on any type of observations to infer the posterior distribution of plausible state trajectories, without retraining. Our unified probabilistic framework flexibly tackles multiple inference tasks – reanalysis, filtering, and forecasting – using the same model, eliminating the need for task-specific architectures or training procedures. Experiments demonstrate physical consistency on a global scale and good reconstructions from observations, while showing competitive forecasting skills. Our results establish latent score-based data assimilation as a promising foundation for future global atmospheric modeling systems.

1 Introduction

Recent advances in deep learning have shown remarkable success in weather forecasting, with neural network-based models achieving accuracy comparable or superior to traditional numerical weather prediction systems [1–6] at a fraction of the computational cost.

The accuracy of any forecast, however, fundamentally depends on having a good estimate of the current atmospheric state. For this reason, weather centers must routinely solve a critical inverse problem: identifying the atmospheric states that are compatible with the vast amounts of observational data they receive from ground stations, satellites, and other sources. For example, the European Center for Medium-Range Weather Forecasts (ECMWF) processes around 60 million quality-controlled observations daily from approximately 90 satellite instruments. Traditionally, weather centers have relied on variational data assimilation methods [7–11] such as 4D-Var, which formulate the problem as finding the maximum a posteriori estimate of the atmospheric state for a window of recent observations [12]. These methods have proven remarkably successful in operational settings but face inherent challenges: (a) they require differentiating through complex physical models, (b) they rely on linear approximations that may not fully capture highly nonlinear atmospheric processes, especially during extreme weather events, and (c) they typically provide point estimates rather than full probabilistic predictions.

Building on score-based data assimilation (SDA) [13, 14], which has demonstrated promising results at regional scales [15, 16], we introduce Appa, a spatio-temporal latent diffusion model that scales SDA to global atmospheric states. By operating in a compressed latent space, Appa efficiently handles the high dimensionality and complex dynamics of planetary-scale weather systems while providing access to the full posterior distribution of atmospheric trajectories. The contributions of this work are

*gandry@uliege.be

- (1) a 500M-parameter autoencoder for **compressing atmospheric states to a low-dimensional latent space** with a $450\times$ compression factor, while preserving realistic reconstructions;
- (2) a 1B-parameter spatio-temporal diffusion model for **sampling latent atmospheric trajectories**;
- (3) a demonstration of Appa’s ability to perform several tasks like reanalysis, filtering and forecasting, producing global atmospheric trajectories at **0.25-degree resolution and 1-hour intervals**, conditioned on diverse observations at inference;
- (4) an evaluation of Appa’s physical consistency, unconditional and conditional generation capacity, and computational efficiency.

The paper is structured as follows. Section 2 provides background on data assimilation and score-based data assimilation. Section 3 describes the Appa framework, covering compression, spatio-temporal diffusion, and conditional sampling. Section 4 reports experimental results on global weather data assimilation tasks. Finally, Section 5 discusses implications, limitations, and future directions.

2 Background

Data assimilation Formally, let $x_{1:L} = (x_1, x_2, \dots, x_L) \in \mathbb{R}^{L \times V \times C}$ denote a trajectory of L atmospheric states represented as C physical fields (or channels) over a mesh of V vertices. Let $p(x_1)$ be the initial state prior and $p(x_{i+1} | x_i)$ be the transition dynamics from state x_i to state x_{i+1} . An observation or set of observations $y \in \mathbb{R}^M$ of the state trajectory $x_{1:L}$ follows an observation process $p(y | x_{1:L})$. This process is generally formulated as $y = \mathcal{M}(x_{1:L}) + \eta$, where the measurement function $\mathcal{M} : \mathbb{R}^{L \times V \times C} \mapsto \mathbb{R}^M$ might be non-linear and $\eta \in \mathbb{R}^M$ is a stochastic additive term representing observational error that accounts for instrumental noise and systematic uncertainties. In this setting, the goal of data assimilation is to solve the inverse problem of inferring plausible trajectories $x_{1:L}$ given an observation y , that is, to estimate the trajectory posterior

$$p(x_{1:L} | y) = \frac{p(y | x_{1:L})}{p(y)} p(x_1) \prod_{i=1}^{L-1} p(x_{i+1} | x_i). \quad (1)$$

Score-based data assimilation Rozet et al. [13] tackle the trajectory posterior inference problem by learning a diffusion model [17–19] of the trajectory prior $p(x_{1:L})$. Adapting the continuous-time formulation of Song et al. [19], trajectory samples $x_{1:L} \sim p(x_{1:L})$ are progressively perturbed through a diffusion process expressed as a stochastic differential equation (SDE)

$$dx_{1:L}(t) = f(t) x_{1:L}(t) dt + g(t) dw(t), \quad (2)$$

where $f(t) \in \mathbb{R}$ is the drift coefficient, $g(t) \in \mathbb{R}_+$ is the diffusion coefficient, $w(t)$ denotes a standard Wiener process and $x_{1:L}(t)$ is the perturbed trajectory at time $t \in [0, 1]$. Because the SDE is linear with respect to $x_{1:L}(t)$, the perturbation kernel from $x_{1:L}$ to $x_{1:L}(t)$ is Gaussian and takes the form

$$p(x_{1:L}(t) | x_{1:L}) = \mathcal{N}(x_{1:L}(t) | \alpha(t) x_{1:L}, \Sigma(t)) \quad (3)$$

where $\alpha(t)$ and $\Sigma(t) = \sigma(t)^2 I$ can be derived analytically from $f(t)$ and $g(t)$ [20, 21]. Crucially, the forward SDE (2) admits a family of reverse SDEs [19–21]

$$dx_{1:L}(t) = \left[f(t) x_{1:L}(t) - \frac{1 + \eta^2}{2} g(t)^2 \nabla_{x_{1:L}(t)} \log p(x_{1:L}(t)) \right] dt + \eta g(t) dw(t), \quad (4)$$

where $\eta \geq 0$ is a parameter controlling stochasticity. This means that we can draw noise samples $x_{1:L}(1) \sim p(x_{1:L}(1)) \approx \mathcal{N}(0, \Sigma(1))$ and gradually remove the noise therein to obtain $x_{1:L}(0) \sim p(x_{1:L}(0)) \approx p(x_{1:L})$ by simulating Eq. (4) from $t = 1$ to 0. In this work, we adopt a variance exploding SDE [22] for which $f(t) = 0$, $\alpha(t) = 1$ and $\sigma(t) = \frac{t}{1-t}$.

In practice, the score function $\nabla_{x_{1:L}(t)} \log p(x_{1:L}(t))$ in Eq. (4) is unknown. Relying on the Markovian properties of dynamical systems, Rozet et al. [13] show that its elements $\nabla_{x_i(t)} \log p(x_{1:L}(t))$ can be approximated locally by $\nabla_{x_i(t)} \log p(x_{i-k:i+k}(t))$ for some $k \geq 1$ and propose to train a score network $s_\phi(x_{i-k:i+k}(t), t)$ over short segments $x_{i-k:i+k}$, called blankets, to approximate the score. At inference, they compose these local scores to approximate the full score and generate arbitrarily long trajectories from the prior $p(x_{1:L}(t))$.

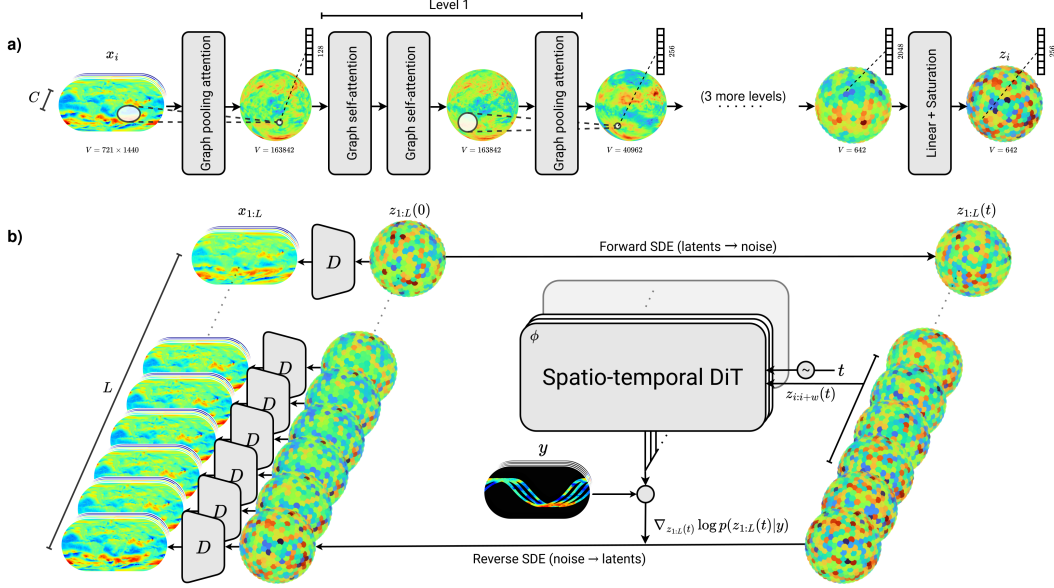


Figure 1. Overview of Appa’s architecture. a) The autoencoder transforms atmospheric states between the original high-dimensional space and the compact latent representation. The encoder E progressively downsamples the input 721×1440 mesh through increasingly coarser icosahedral meshes (left to right), while expanding channel dimensions from 71 to 2048 before projecting to 256 latent channels. The decoder D (not shown) mirrors this architecture to reconstruct the original atmospheric states. b) The latent diffusion process operates entirely in the compressed space, where latent trajectories $z_{1:L}(0)$ are perturbed via a forward SDE to create noisy trajectories $z_{1:L}(t)$. The spatio-temporal DiT denoiser learns to reverse this process, computing trajectory scores that enable sampling from the distribution. At generation, observations y can be incorporated through the score decomposition to generate posterior trajectories matching observed data.

Finally, to generate from the posterior $p(x_{1:L} | y)$, one needs to approximate the posterior score

$$\nabla_{x_{1:L}(t)} \log p(x_{1:L}(t) | y) = \nabla_{x_{1:L}(t)} \log p(x_{1:L}(t)) + \nabla_{x_{1:L}(t)} \log p(y | x_{1:L}(t)), \quad (5)$$

and plug it into the reverse SDE (4). Remarkably, the likelihood score $\nabla_{x_{1:L}(t)} \log p(y | x_{1:L}(t))$ in Eq. (5) can be approximated without any retraining under moderate assumptions on the observation process $p(y | x_{1:L})$ [13, 23–25].

3 Appa

In the case of global weather data assimilation, atmospheric states $x_i \in \mathbb{R}^{V \times C}$ can be very large, making the training of a score network and sampling from the posterior $p(x_{1:L} | y)$ computationally infeasible. For example, at ERA5 resolution, a 1-week hourly trajectory $x_{1:L}$ would have a $L = 7 \times 24$ length, $V = 721 \times 1440$ vertices and $C \in \mathcal{O}(100)$ channels, for a total dimension $L \times V \times C$ to the order of 10^{11} . To address this computational challenge, recent works within the image and video generation literature [26–28] consider generating in the latent space of an autoencoder instead of the original pixel space. In the following, we present how we adapt this approach within the score-based data assimilation [13] framework. The result, called Appa, is a 1.5B-parameter spatio-temporal latent diffusion model trained on ERA5 reanalysis data [29].

3.1 Compressing atmospheric states

The first component of Appa is a 500M-parameter encoder-decoder pair (E_ψ, D_ψ) that compresses high-dimensional atmospheric states x_i into lower-dimensional Gaussian latent representations with mean $\mathbb{E}[z_i | x_i] = E_\psi(x_i)$ and fixed variance σ_z^2 , as illustrated in Figure 1a. The latent states z_i are orders of magnitude smaller than the atmospheric states x_i , allowing for inference in the latent space

at a much lower computational cost. After inference, the generated latent states are mapped to the original space via the decoder $\hat{x}_i = D_\psi(z_i)$.

Architecture Unlike typical image autoencoders that operate on regular grids, our architecture operates on geodesic meshes to handle the spherical geometry of global weather data. The encoder E_ψ processes the input 721×1440 latitude-longitude grid (about 28 km resolution at the equator) through a series of increasingly coarser icosahedral meshes, from a 7-times refined icosahedron (163842 vertices) down to a 3-times refined icosahedron (642 vertices) for the latent representation. As we progress through the series of meshes, the number of channels per vertex increases geometrically from 71 input channels to 128, 256, 512, 1024 and finally 2048 channels at the coarsest level, thereby compensating the spatial coarsening with richer representations. A final projection passes from 2048 to 256 channels per vertex, creating a compact latent representation $z \in \mathbb{R}^{642 \times 256}$ with an overall compression factor of 450 with respect to the input. Each level of the encoder consists of (i) a graph pooling attention layer [30–32] that maps features from the finer mesh at the previous level to the coarser mesh at the current level and (ii) multiple graph local self-attention blocks [4, 32, 33] that combine vertex features while preserving the spherical topology. Graph pooling operations are restricted to 1-hop neighborhoods of the finer grids while self-attention blocks operate on a 3-hop neighborhoods on the icosahedral meshes (with 1 hop corresponding to a geodesic distance of 66 km at the finest resolution and 1050 km at the coarsest resolution), ensuring computational efficiency while preserving local spatial relationships. To prevent the encoder from using latents with large amplitudes, we use a saturating function of the form $z \mapsto z/\sqrt{1+z^2/25}$ at its output. We find using a saturation to be simpler and more effective than the typical KL regularization in variational autoencoders [34]. The decoder D_ψ mirrors the encoder architecture with graph upsampling layers and self-attention blocks that progressively refine the latent representation back to the original mesh resolution. Finally, auxiliary context information such as solar irradiance, spherical coordinates and time of year, is provided both as additional input channels to the encoder and per-vertex sinusoidal embeddings [35, 36]. Further details can be found in Appendix B.1.

Training The full architecture is trained end-to-end by minimizing a latitude-weighted mean squared error [3] between input states and their reconstructions. During training, we fix the latent noise variance to $\sigma_z^2 = 10^{-2}$. The latter helps improve robustness and prevents overfitting.

3.2 Latent diffusion of atmospheric trajectories

The second component of Appa is a 1B-parameter spatio-temporal diffusion model operating in latent space, as illustrated in Figure 1b. Like in the original SDA [13] framework, the diffusion model is parameterized by a local score network $s_\phi(z_{i:i+w}(t), t)$ trained with latent blankets $z_{i:i+w}$ of size w to approximate the score $\nabla_{z_{i:i+w}(t)} \log p(z_{i:i+w}(t))$. During inference, the latent trajectory $z_{1:L}(t)$ is split into $L-w/\Delta + 1$ blankets $z_{i:i+w}$ for $i = 1 + n\Delta$, where $\Delta \leq w$ is the stride between blankets. Then, the local score network is applied to each blanket and Δ elements are taken from each local score and recombined into an approximation of the full prior score $\nabla_{z_{1:L}(t)} \log p(z_{1:L}(t))$. The full procedure is described in Algorithm 2.

Algorithm 1 Training $s_\phi(z_{i:i+w}(t), t)$

```

1 for  $i = 1$  to  $N$  do
2    $x_{1:L} \sim p(x_{1:L})$ 
3    $i \sim \mathcal{U}(\{1, \dots, L - w\})$ 
4    $t \sim \mathcal{U}(0, 1), \epsilon \sim \mathcal{N}(0, I)$ 
5   for  $j = i$  to  $i + w$  do
6      $z_j \leftarrow E_\psi(x_j)$ 
7    $z_{i:i+w}(t) \leftarrow \alpha(t) z_{i:i+w} + \sigma(t) \epsilon$ 
8    $\ell \leftarrow \text{SCOREMATCHING}(s_\phi, z_{i:i+w}(t), t, \epsilon)$ 
9    $\phi \leftarrow \text{GRADIENTDESCENT}(\phi, \nabla_\phi \ell)$ 
```

Algorithm 2 Composing $s_\phi(z_{i:i+w}(t), t)$

```

1 function  $s_\phi(z_{1:L}(t), t)$ 
2    $a \leftarrow (w - \Delta)/2$ 
3    $b \leftarrow a + \Delta$ 
4    $s_{1:a} \leftarrow s_\phi(z_{1:1+w}(t), t)[a]$ 
5   for  $n = 0$  to  $(L - w)/\Delta + 1$  do
6      $i \leftarrow 1 + n\Delta$ 
7      $s_{i+a:i+b} \leftarrow s_\phi(z_{i:i+w}(t), t)[a:b]$ 
8    $s_{L-w+b:L} \leftarrow s_\phi(z_{L-w:L}(t), t)[b:]$ 
9   return  $s_{1:L}$ 
```

In practice, we find that blankets covering $w = 25$ hours provide a good trade-off between capturing long-term atmospheric dynamics and maintaining reasonable computational requirements. In the original SDA [13] framework, the stride Δ was implicitly set to 1. We find in our experiments that

using a larger stride $\Delta = 13$ does not measurably affect temporal consistency, while greatly reducing the number of network evaluations.

Architecture The score network adapts the DiT [27] architecture to operate on spatio-temporal graphs, consisting of transformer blocks with multi-head self-attention and feedforward networks modulated by the diffusion time. Each attention block processes the full spatio-temporal graph by attending jointly to all tokens across all timesteps, effectively capturing both intra-mesh spatial relationships and inter-mesh temporal dependencies in a unified attention mechanism. As in the autoencoder, additional context information such as spherical coordinates and time of year, is provided as per-vertex sinusoidal embedding. Further details on the architecture can be found in Appendix B.2.

Training In practice, we do not learn the score function directly. We follow the framework of Karras et al. [37] and train a denoiser model to estimate $\mathbb{E}[z_{1:L} \mid z_{1:L}(t)]$ which is directly linked to the score function through the first order Tweedie’s formula.

3.3 Sampling conditionally on weather observations

The third and last component of Appa is a posterior sampling algorithm for conditioning on atmospheric observations. In our case, because we operate in latent space, the decomposition in Eq. (5) becomes

$$\nabla_{z_{1:L}(t)} \log p(z_{1:L}(t) \mid y) = \nabla_{z_{1:L}(t)} \log p(z_{1:L}(t)) + \nabla_{z_{1:L}(t)} \log p(y \mid z_{1:L}(t)). \quad (6)$$

The key challenge is that the observation process $p(y \mid x_{1:L})$ is defined for atmospheric states x_i rather than latent states z_i . Assuming we have a series of observations $y_{1:L}$ of the form $y_i = \mathcal{M}_i(x_i) + \eta_i$, our approach approximates the process from z_i to y_i as the combination of the decoder D_ψ and measurement \mathcal{M}_i . Formally, we model the observation process as

$$p(y_{1:L} \mid z_{1:L}) \approx \mathcal{N}(y_{1:L} \mid \mathcal{A}(z_{1:L}), \Sigma_y), \quad (7)$$

where $\mathcal{A}(z_{1:L}) = (\mathcal{M}_1(D_\psi(z_1)) \dots \mathcal{M}_L(D_\psi(z_L)))^\top$ and Σ_y is the variance of the stochastic terms η_i . We emphasize that Eq. (7) is an approximation as the decoder D_ψ does not reconstruct atmospheric states perfectly. Nevertheless, it allows us to apply off-the-shelf posterior sampling methods for conditioning with respect to observations, as in the original SDA [13] framework.

In this work, we use the moment matching posterior sampling (MMPS) method recently proposed by Rozet et al. [24], which assumes the approximation

$$p(y \mid z_{1:L}(t)) \approx \mathcal{N}(y \mid \mathcal{A}(\mathbb{E}[z_{1:L} \mid z_{1:L}(t)]), \Sigma_y + A \mathbb{V}[z_{1:L} \mid z_{1:L}(t)] A^\top). \quad (8)$$

We note that because our forward operator \mathcal{A} is non-linear, we instead use its Jacobian A in the covariance. We refer the reader to Rozet et al. [24] for details on how to produce the likelihood score $\nabla_{z_{1:L}(t)} \log p(y \mid z_{1:L}(t))$ from Eq. (8).

Once the prior score is trained, Appa can assimilate any type of observations expressible as a measurement $\mathcal{M}(x_{1:L})$ of the atmospheric state trajectory $x_{1:L}$, including satellite radiance, weather station measurements, or radar data, without using these observations during training. This flexibility makes Appa suitable for operational settings where observation networks constantly evolve and new instruments are regularly deployed.

4 Experiments

4.1 Protocol and hardware

We train and evaluate Appa on ERA5 reanalysis data [29] spanning from January 1999 to December 2019. We follow standard chronological splitting: 1999–2017 for training, 2018 for validation, and 2019 for testing. We use 6 surface and 5 atmospheric variables across 13 pressure levels for a total of $C = 71$ physical fields, with top-of-atmosphere solar radiation as a context variable to inform the model about diurnal and seasonal cycles. We refer to Appendix A for more details about the data.

The complete model is composed of a 527M-parameter autoencoder and a 967M-parameter denoiser. Training was performed on 64 A100 40GB GPUs, with the autoencoder trained for 2 days and the denoiser for 5 days. Architectures and hyperparameters are detailed in Appendix B.

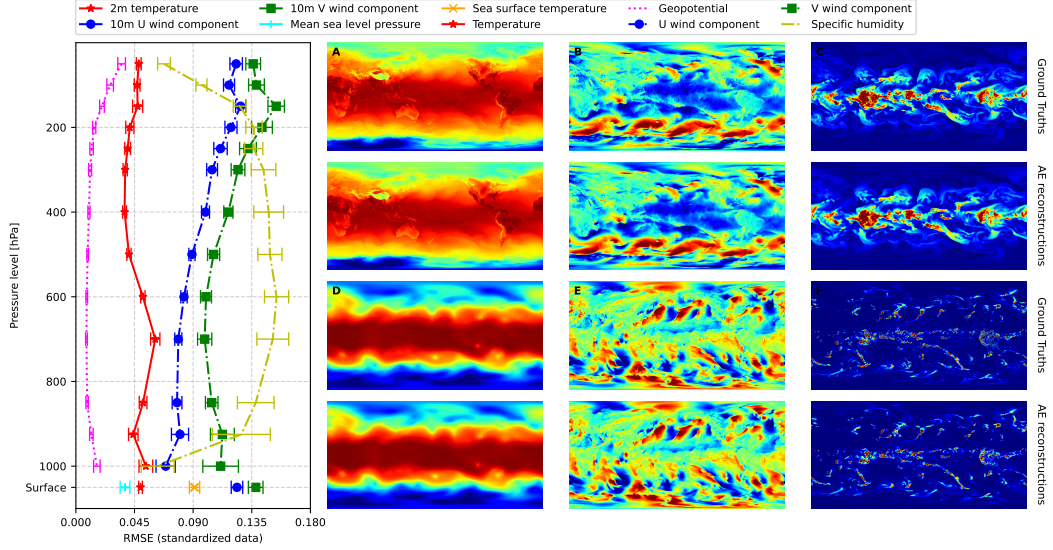


Figure 2. Autoencoder reconstruction quality. (Left) Root mean square error (RMSE) for reconstructing standardized surface and atmospheric variables across pressure levels. Lower values indicate better reconstruction quality. The autoencoder performs well overall, with particularly low errors for temperature and geopotential, which are easier to predict low-frequency variables. In contrast, the wind components and specific humidity, which exhibit higher variability, have slightly larger errors. (Right) Visual comparison between ground-truth fields and reconstructed fields on 2019-04-24 20:00 UTC for selected variables and pressure levels (resp., the 2m temperature in **A**, the 10m U wind component in **B**, the specific humidity at 300 hPa in **C**, the geopotential at 300 hPa in **D**, the 10m V wind component in **E** and the total precipitation in **F**), demonstrating the ability of the autoencoder to preserve key spatial patterns and features despite the $450\times$ compression factor.

4.2 Latent representations

We first evaluate the quality of the latent space learned by the autoencoder, focusing on its ability to preserve the main physical properties of atmospheric states despite the $450\times$ compression factor.

Figure 2 shows the root mean square error (RMSE) of reconstructed standardized variables across different pressure levels. The reconstruction errors show varying performance across variables, with approximately half of the measurements achieving errors below 0.10, demonstrating the autoencoder’s ability to represent global atmospheric states with good fidelity. Temperature and geopotential variables are particularly well reconstructed, while specific humidity and wind components exhibit slightly higher errors, likely due to their inherent variability and chaotic nature. The distribution of signed errors in Figure 3 reveals that reconstruction errors are centered around zero, indicating that the autoencoder does not introduce significant biases in the reconstructed fields. These errors are also symmetrically distributed, suggesting that the autoencoder captures the variability of the atmospheric state without introducing systematic distortions. Importantly, the narrow and peaked error distributions show that the vast majority of grid points are reconstructed with high accuracy, with only a small fraction exhibiting significant deviations from the original values.

Beyond point-wise accuracy, we evaluate the preservation of physical consistency by analyzing the power spectra of original and reconstructed fields, as shown in Figure 4. The spectra show excellent agreement for large to medium spatial scales, with a slight discrepancy at the smallest scales. This expected discrepancy reflects the autoencoder’s progressive downsampling architecture described in Section 3.1, which naturally prioritizes capturing the dominant modes of variability typically associated with larger scales while achieving significant dimensionality reduction. Importantly, this approach successfully preserves the vast majority of the energy in the system, as atmospheric energy is predominantly concentrated in larger scales, with energy levels dropping by several orders of magnitude at the smallest scales where the discrepancies appear.

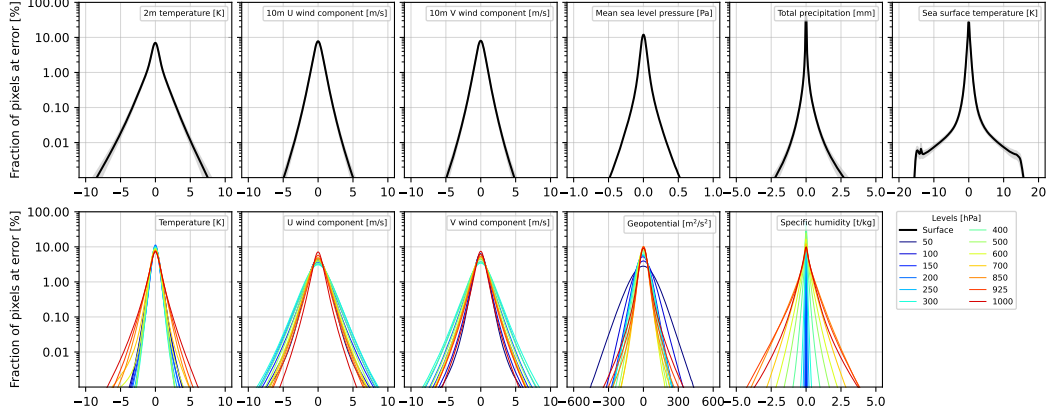


Figure 3. Autoencoder error distribution analysis. Signed reconstruction errors for surface variables (top) across all grid points and atmospheric variables (bottom) across all pressure levels. Shaded gray area for surface variables corresponds to error spread. In both cases, the concentrated distributions centered around zero demonstrate unbiased and precise predictions. Given $721 \times 1440 = 1,038,240$ grid points, a 0.01% fraction on the y-axis corresponds to approximately 100 grid points, indicating that large errors are rare.

Compared to previous studies on neural compression of atmospheric data [38], our autoencoder achieves better or comparable reconstruction errors. Overall, these results demonstrate that our autoencoder effectively compresses atmospheric states into a low-dimensional latent space while preserving the main characteristics of the data.

4.3 Unconditional generation of atmospheric trajectories

We now assess the quality of atmospheric trajectories generated by our latent diffusion model. As shown in Figure 4, the power spectra of generated samples closely follow those of autoencoder reconstructions across most spatial scales, indicating that the diffusion model successfully learns the distribution in latent space. Like the autoencoder reconstructions, diffusion-generated samples accurately preserve energy distributions from large to medium scales and show the same drop-off at higher frequencies (smaller scales). This consistency confirms that our latent diffusion model effectively captures the statistical properties available in the compressed representation, without introducing additional spectral distortions.

To further evaluate physical consistency, we examine whether our model preserves important physical relationships between variables. First, we analyze the consistency between two different estimators of altitude at given pressure levels. Using the geopotential Φ , altitude can be derived as

$$H = \frac{\Phi R_e}{g_0 R_e - \Phi} \quad (9)$$

where R_e is Earth’s radius and g_0 is the Earth gravitational acceleration at the surface. Alternatively, the equation below (which relies on the ideal gas law and hydrostatic equation) relates altitude to pressure and temperature as

$$\log \frac{p_0}{p_H} = \frac{M g_0}{R} \int_0^H \frac{1}{T_h} \partial h, \quad (10)$$

where R is the universal gas constant, M is an approximation of the atmosphere’s molar mass, p_h, T_h are pressure and temperature at height h , and p_0, T_0 are the theoretical pressure and temperature at sea level. This integral can be approximated to extract H using several assumptions about the temperature profile, as detailed in Appendix C. When comparing these two estimators, Figure 5 shows that our generated samples maintain the same systematic differences ΔH as seen in ground-truth data, with discrepancies in the order of $\mathcal{O}(10m)$ affecting only a tiny portion of the samples. This remarkable consistency indicates that our model successfully preserves the physical relationships between temperature, pressure, and geopotential, allowing altitude to be estimated through two independent methods with nearly identical accuracy to the original ERA5 data.

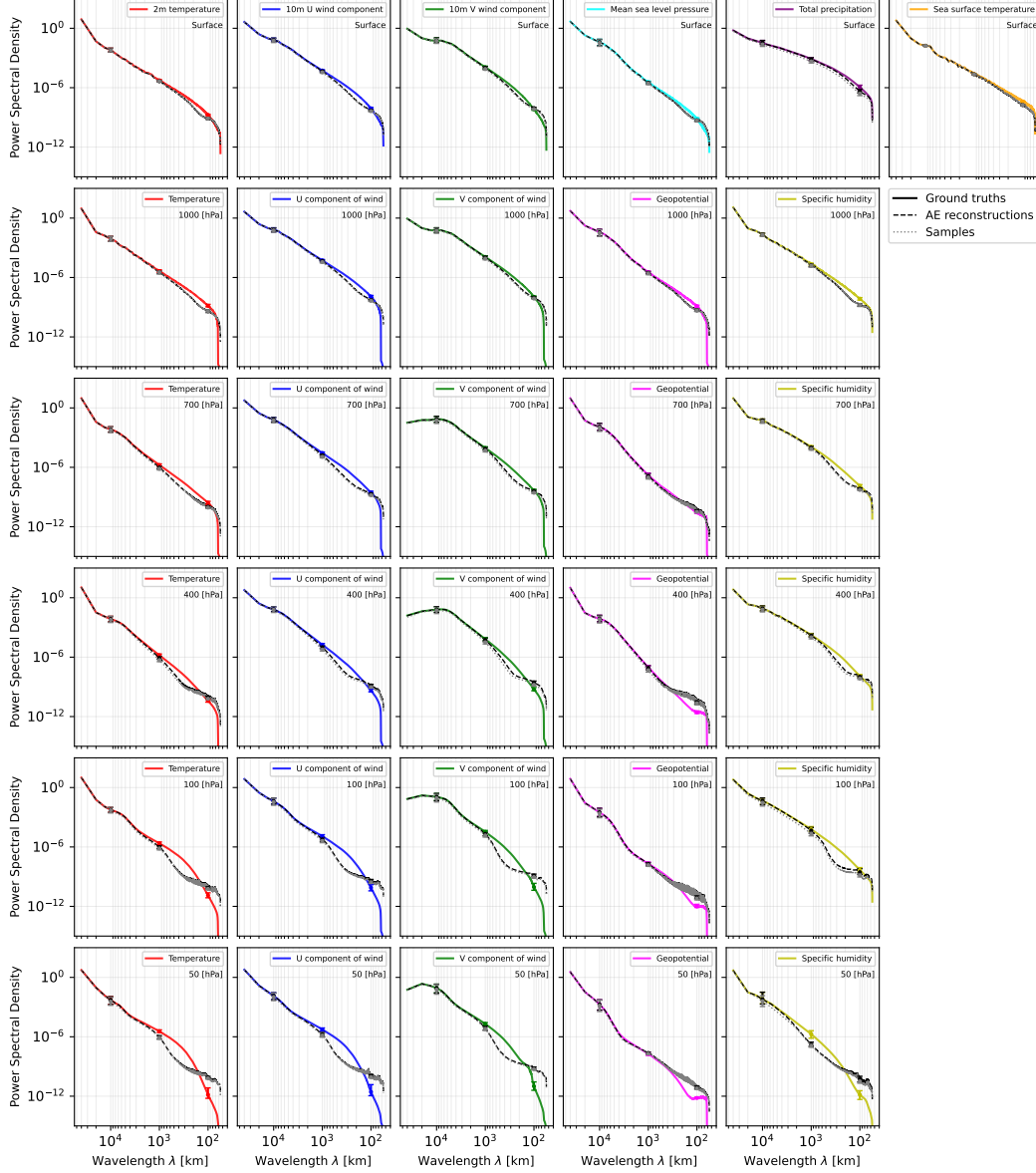


Figure 4. Spectral analysis. (Top row) Power spectral density of ground truth, autoencoder reconstructions, and unconditional diffusion samples across wavelengths for surface variables. (Lower rows) Power spectral density across wavelengths for atmospheric variables at selected pressure levels. Lines show median values and error bars indicate the 5th to 95th percentiles. The close alignment between the curves demonstrates that both the autoencoder and the diffusion model preserve the energy distribution across most spatial scales. Deviations begin to appear at wavelengths around 1000km, which corresponds to roughly 40 grid cells at our 0.25-degree resolution at the equator. These differences become more pronounced at smaller scales, suggesting that while large-scale atmospheric patterns are well-preserved, features spanning fewer than 40 grid cells show some energy loss in the compression and generation processes.

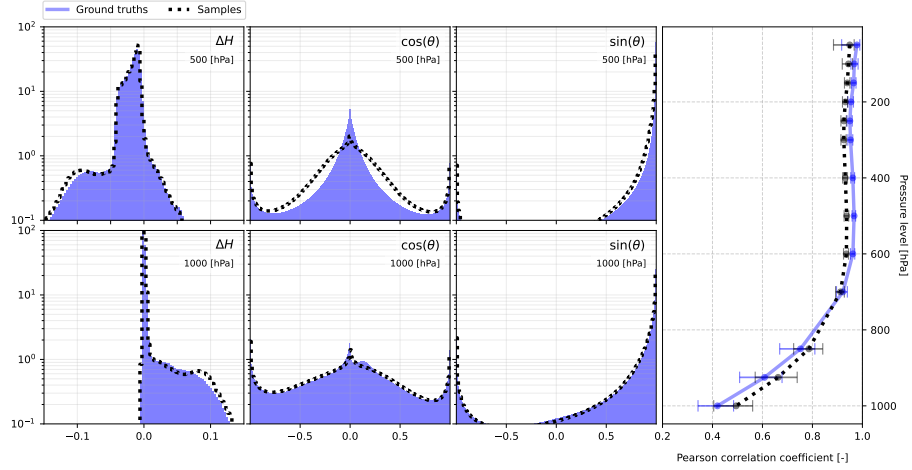


Figure 5. Physical consistency analysis of generated atmospheric states. (Top row) Analysis of altitude consistency at 500 hPa showing the difference ΔH between two independent altitude estimators, and geostrophic balance assessment through the cosine and sine of the angle θ between wind direction and geopotential gradients, demonstrating angles concentrated around 90° . (Bottom row) Same metrics at 1000 hPa demonstrating the presence of a significant ageostrophic component near the surface. (Right) Correlation coefficient between wind magnitude and geopotential gradient magnitude across pressure levels, showing strong correlation at upper levels (near 1) with a consistent decrease toward the surface in both ERA5 data (blue) and generated samples (black dots), confirming Appa’s ability to capture complex physical relationships.

Second, we examine geostrophic balance, the theoretical equilibrium between pressure gradient forces and Coriolis forces that governs large-scale atmospheric motion. In this balance, in the absence of vertical motion, friction, and isobaric curvature, wind direction should be perpendicular to geopotential gradients, with wind speed proportional to gradient magnitude. This relationship can be expressed by comparing two quantities: (1) the angle θ between wind and geopotential gradients, which should approach 90° in geostrophic conditions, and (2) the correlation between wind speed magnitude and geopotential gradient magnitude, which should approach 1 in perfect geostrophic balance. Figure 5 shows that our generated samples accurately reproduce both aspects of this relationship. At 500 hPa, the approximate level of non-divergence with minimal surface friction effects, both ERA5 data and our generated samples show angles concentrated around 90° . Near the surface at 1000 hPa, where additional forces become significant, both datasets show a systematic deviation in angle. Similarly, the correlation between wind speed and geopotential gradient magnitudes in our samples closely matches the patterns observed in ERA5 data, exhibiting near-perfect correlation (close to 1) at lower pressure levels and following the same decreasing trend as pressure increases toward the surface, where ageostrophic components become more prominent.

These results demonstrate that our latent diffusion model not only preserves the statistical properties of atmospheric fields but also maintains important physical relationships between variables producing trajectories that are physically consistent and realistic. While these analyses confirm strong spatial consistency and physical fidelity, future work should extend our evaluation to more thoroughly quantify the temporal consistency of generated trajectories.

Finally, from a computational perspective, generating a blanket of size $w = 25$ takes barely 20 seconds on a single GPU, with 64 network evaluations in the sampler. The generation of longer trajectories scales efficiently with hardware availability, as multiple GPUs can process individual blankets in parallel with minimal synchronization overhead, enabling the rapid generation of extended atmospheric sequences. We provide supplementary results for long trajectory generation in Appendix D.

4.4 Assimilation of weather observations

To evaluate Appa’s flexibility for assimilation tasks, we study several scenarios: reanalysis, filtering, observational forecasting, and full-state forecasting. For the three first tasks, we assimilate ground-station observations for all six surface variables, alongside simulated satellite scans of 65 atmospheric variables. The station network consists of approximately 11,000 real-world measurement locations [39], covering roughly 1% of the grid points. Ground stations are sparse and globally distributed, whereas simulated satellite orbital paths provide dense spatial observations but with restricted temporal and spatial reach.

For the purpose of the demonstration, we model observations as Gaussian distributions centered on the ERA5 ground truth, with standard deviations of 1% for ground stations and 10% for satellite measurements. While this simplified observation model does not capture the full complexity of real-world instrumental errors and biases, it provides a suitable testbed for demonstrating Appa’s capabilities in handling multi-source observations with varying reliability levels. More sophisticated observation models could be incorporated in operational settings.

Like for unconditional generation, blankets can be distributed across multiple GPUs, making trajectory sampling embarrassingly parallel and total time nearly constant regardless of the number of blankets, aside from negligible communication overhead. On one GPU, a single blanket conditioned entirely on observations (for reanalysis or observational forecasting) takes about 1 hour and 15 minutes, while conditioning on latent space (full-state forecasting) completes in just 7 minutes. These timings correspond to 32 predictor-corrector steps [19], 2 corrector steps, 2 MMPS iterations, and a blanket size $w = 25$. Unlike all-at-once parallel generation, autoregressive forecasting requires the sequential generation of blankets, resulting in a total computation time of approximately $L/m \times 7$ minutes to forecast L hours with m hours per autoregressive step.

Reanalysis In Earth system science, reanalysis plays a central role and supports a wide range of downstream applications from climate monitoring to policy-making. Reanalysis aims to fill gaps in historical observational records by reconstructing a trajectory of the atmospheric state that is consistent in space and time. From a mathematical perspective, reanalysis reduces to the problem of sampling

$$x_{1:L} \sim p(x_{1:L} \mid y_{1:L}), \quad (11)$$

a special case of Eq. (1) for which the observation y is a sequence of historical measurements from satellites, ground stations, and other sensors over a finite time window. Interestingly, the statistical problem of filtering

$$x_L \sim p(x_L \mid y_{1:L}) = \int p(x_L, x_{1:L-1} \mid y_{1:L}) dx_{1:L-1} \quad (12)$$

can be formulated trivially as a reanalysis sub-task by marginalizing (11) over previous states. In practice, this means that we can extract x_L from samples of the full posterior distribution $p(x_{1:L} \mid y_{1:L})$ to obtain valid samples from the filtering distribution $p(x_L \mid y_{1:L})$, all with the same model.

As demonstrated in Figures 6 and 7, Appa successfully reconstructs atmospheric states that closely match the ground truth while respecting observational constraints, despite having access to only a small fraction of the full state. Moreover, unlike traditional approaches that typically process short intervals, Appa scales to windows spanning several days, as demonstrated here for a 1-week assimilation window. The figures also illustrate Appa’s inherent ensemble generation capability. By sampling multiple times from the posterior distribution, Appa can produce diverse yet physically consistent trajectories that all honor observational constraints.

Figure 8 quantifies the performance of both reanalysis and filtering tasks by comparing posterior samples against the ground-truth ERA5 states used to generate the synthetic observations $y_{1:L}$. As expected, conditioning on more observations by extending the assimilation window reduces both the Skill score (RMSE of the posterior mean) and the Continuous Ranked Probability Score (CRPS), indicating improved performance. However, the improvements eventually saturate, particularly after 24 hours of observations. This saturation can be attributed to several factors: (a) posterior inference does not guarantee exact reconstructions as the learned prior steers samples toward plausible atmospheric states, sometimes away from observations; (b) the assumed noise in the observation model limits precision; (c) the linearization of the observation model introduces approximation errors; and (d) imperfections in the autoencoder reconstruction constrain performance. Further

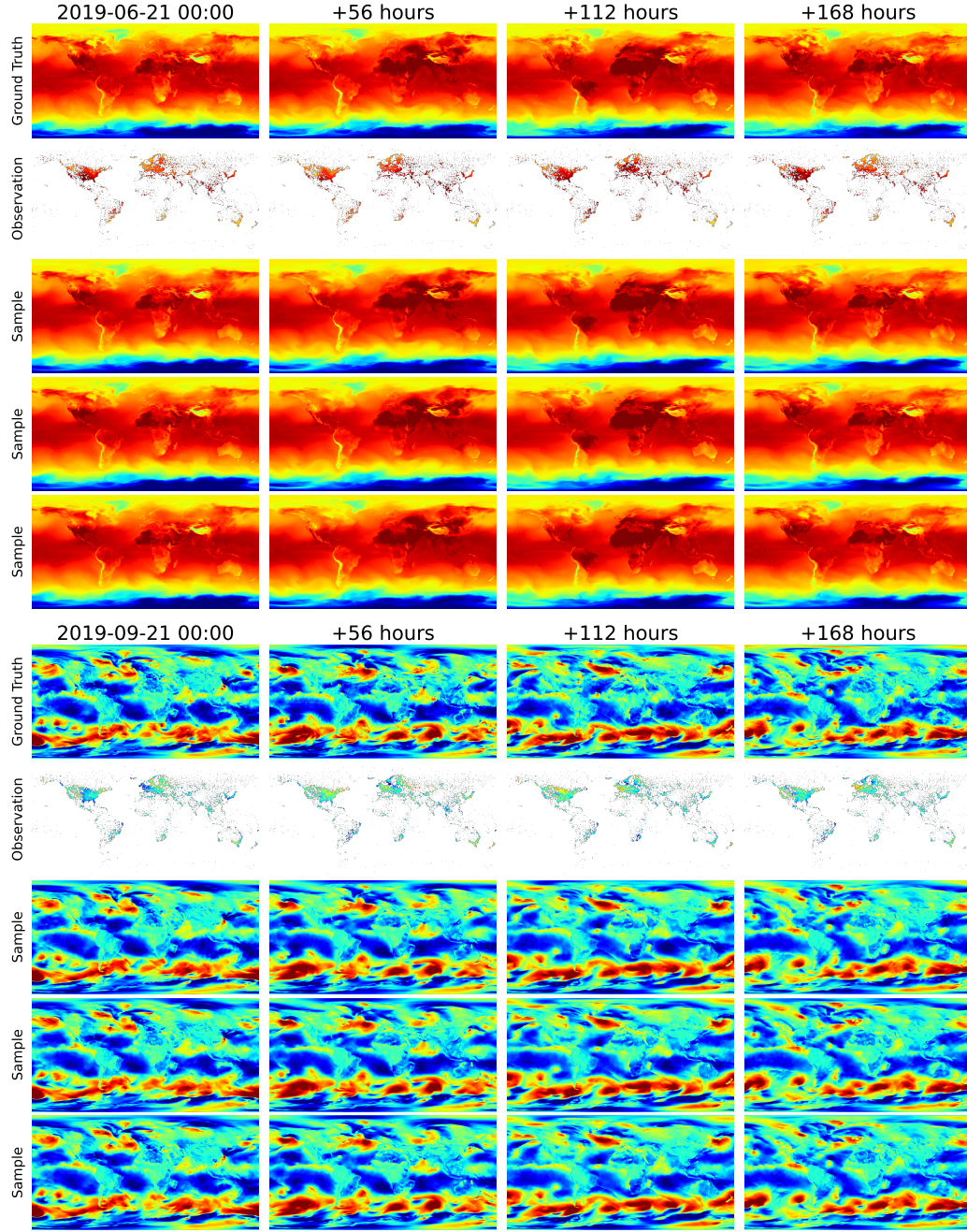


Figure 6. Reanalysis of atmospheric trajectories from sparse observations. (Top) Reconstructed trajectories for 2m temperature during summer, showing ground truth, observation points, and three posterior samples across a one-week period with hourly resolution. (Bottom) Similar visualization for 10m v-component of wind during autumn. In both cases, the posterior samples closely match the ground-truth patterns and features, despite being conditioned on observations covering only approximately 5% of the grid points.

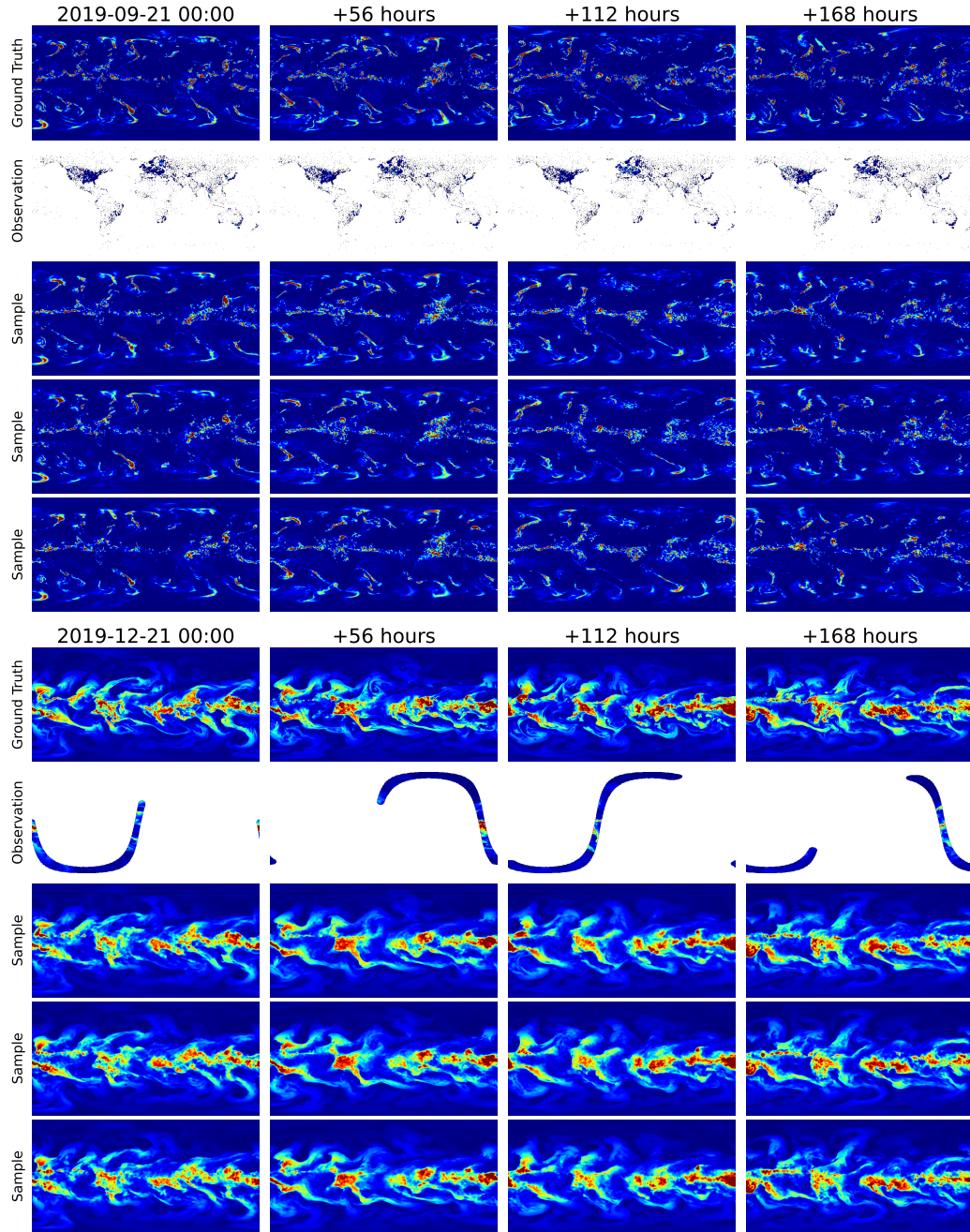


Figure 7. Reanalysis of atmospheric trajectories, similarly to Figure 6. (Top) Reconstructed trajectories for total precipitation during autumn, comparing ground-truth states with posterior samples generated from sparse ground-station observations. (Bottom) Reconstructed trajectories for specific humidity at 300 hPa during winter, with observations derived from simulated satellite coverage.

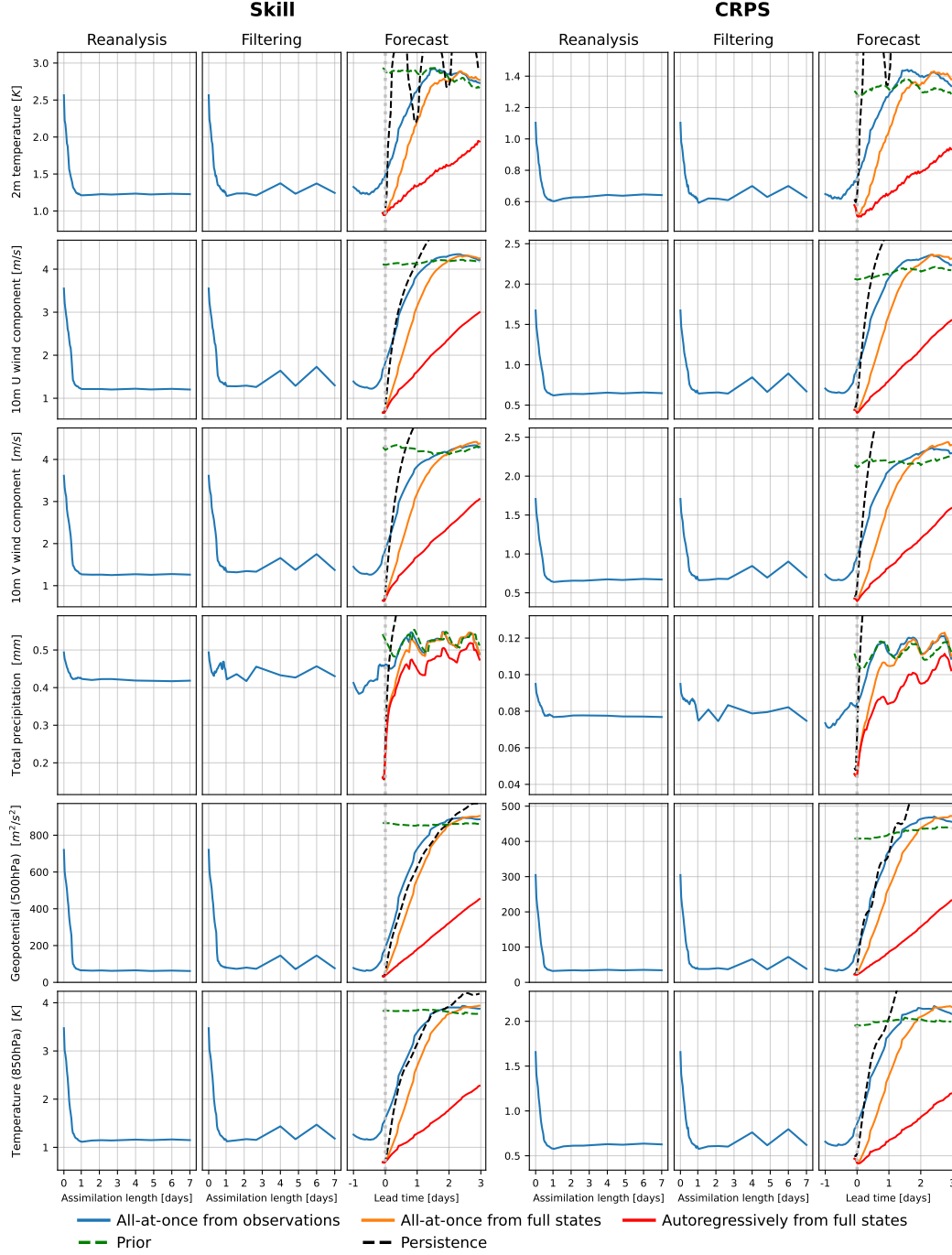


Figure 8. Quantitative evaluation across reanalysis, filtering, and forecasting tasks. (Left) Skill score and (Right) Continuous Ranked Probability Score (CRPS) for six key variables. For reanalysis, values represent averages across all time steps within each assimilation window, while filtering values show only the final time step of each window. Both metrics improve (decrease) as assimilation window size increases, with performance saturating after approximately 24 hours of observations. Filtering performance closely follows reanalysis trends but with slightly higher variability. Within the assimilation window, all-at-once forecasting (blue/orange) achieves performance comparable to reanalysis or filtering (blue), as expected. Beyond this window, the metrics quickly deteriorate and become comparable to unconditional samples (green), regardless of being conditioned on full states or partial observations. In contrast, autoregressive forecasting (red) demonstrates significantly better performance.

results presented in Figure 11 in Appendix D reveal, however, that the ensemble spread continues to decrease as the assimilation window increases, reaching saturation later than observed for Skill and CRPS which indicates that information propagates through the assimilation process, even beyond the model’s window length.

Forecasting Another scenario that we evaluate is *observational* forecasting. In this setting, past observations initialize a forecasting model that predicts future atmospheric states. While traditional approaches separate the past states reanalysis and forecasting steps, Appa can integrate both in a unified probabilistic framework by framing the task as

$$x_{1:L+M} \sim p(x_{1:L+M} \mid y_{1:L}), \quad (13)$$

where $y_{1:L}$ are observations over an assimilation window of length L and M denotes the lead time. This formulation naturally extends the reanalysis scenario with M additional unobserved future states, highlighting the flexibility of our method. The length of the assimilation window L as well as the lead time M can be freely chosen at inference time, without retraining or architectural changes.

Beyond observational forecasting, Appa also supports *full-state* forecasting setups commonly considered in the literature [2, 4, 40, 41], where the model is initialized from one or several known full ground-truth states rather than from partial observations. In our framework, this corresponds to sampling

$$x_{1:L+M} \sim p(x_{1:L+M} \mid x_{1:L}), \quad (14)$$

where the initial segment $x_{1:L}$ is provided by an external source treated as ground truth. In practice, we perform full-state forecasting fully in latent space by encoding past states $z_{1:L} = (E_\psi(x_1) \dots E_\psi(x_L))^T$, generating a trajectory in the latent space conditioned on those encoded states $z_{1:L+M} \sim p(z_{1:L+M} \mid z_{1:L})$, and then finally decoding it back to the state space $x_{1:L+M} = (D_\psi(z_1) \dots D_\psi(z_{L+M}))^T$. This scenario bypasses the use of a decoder during sampling, resulting in a more time-efficient generation.

As discussed by Shysheya et al. [42], all-at-once generation of trajectories performs best when observations are evenly distributed over time. In our case, forecasting inherently creates an asymmetric conditioning problem where all observations lie in the past, making information propagation into future states more challenging. Appa’s flexible architecture, however, also enables autoregressive generation: states can be generated in m -hour increments (in our experiments, $m = 6$ hours) by producing new blankets of size $w = 25$ conditioned on a sliding window that initially contains only ground-truth states and gradually incorporates previously generated states as the forecast advances.

Experimental results shown in Figure 8 demonstrate that all-at-once observational forecasts (13) (blue) begin at performance levels comparable to reanalysis (in terms of Skill and CRPS) within the assimilation window, confirming the consistency of our unified framework. In contrast, all-at-once full-state forecasts (14) (orange) initially show better performance (lower Skill and CRPS scores), reflecting the advantage of conditioning on exact states rather than their posterior estimates derived from sparse observations. However, the quality of both forecasts, regardless of initialization method, deteriorates with lead time and saturates around 48 hours. This suggests that the information from past states quickly vanishes and that forecasts revert to prior samples performance (green) beyond this horizon.

Full-state forecasts reveal striking differences depending on the sampling approach: autoregressive generation (red) demonstrates competitive performance with a gradual, nearly linear degradation over time, while all-at-once sampling (orange) performs notably worse, with even short-term accuracy compromised by distant future predictions. Despite being theoretically equivalent and sharing the same underlying model architecture without task-specific adaptations, these divergent results highlight how the sampling strategy profoundly impacts performance. The autoregressive approach preserves causality at the cost of sequential computation, but offers flexibility through our window mechanism that balances speed and quality by adjusting the ratio of past context to future predictions. Meanwhile, all-at-once generation provides parallelization advantages despite its accuracy limitations. This performance gap in the absence of architectural differences suggests that targeted improvements to sampling mechanisms and conditioning strategies could significantly enhance all-at-once forecasting while preserving its computational benefits.

Overall, Appa demonstrates remarkable flexibility in handling various forecasting scenarios within a single model architecture. Whether performing observational forecasting that seamlessly extends

from reanalysis, full-state forecasting commonly used in benchmarks, or leveraging autoregressive sampling for improved accuracy, Appa adapts to different operational needs without requiring specialized training or architectural modifications. The stronger performance of autoregressive full-state forecasting, in particular, highlights the potential of our approach for operational applications. As we continue to refine the sampling strategies and conditioning mechanisms, we expect to further bridge the gap between general-purpose data assimilation models and specialized forecasting systems, ultimately working toward a unified framework that excels across the entire spectrum of weather prediction tasks.

5 Discussion

Summary We introduce Appa, a latent score-based data assimilation framework that produces global atmospheric trajectories by operating in a compressed latent space. Our approach consists of two key components: a 500M-parameter autoencoder that achieves a $450\times$ compression of atmospheric states, and a 1B-parameter spatio-temporal diffusion model that generates physically consistent trajectories. Appa can be conditioned on various types of observations without retraining, providing access to the full posterior distribution of compatible state trajectories. Our experiments demonstrate that Appa effectively preserves important physical properties of the atmosphere across spatial scales and maintains consistency between interrelated variables. The framework shows strong performance in reanalysis tasks, where it excels at reconstructing complete atmospheric trajectories from sparse observations. While its pure forecasting skill remains slightly inferior to specialized forecasting models, Appa’s unified ability to perform reanalysis, filtering, and observational forecasting demonstrates its versatility across a wide range of weather modeling scenarios.

Related work Weather data assimilation has historically been dominated by variational methods, such as 4D-Var [7, 8, 43], and ensemble Kalman filters [44, 45]. Recently, several approaches have begun to explore the potential of deep learning for this task, as recently reviewed in [46].

Close to our work, DiffDA [47] builds on GraphCast [3] and diffusion models for probabilistic data assimilation. FuXi-DA [48] extends the FuXi [2] weather model with a deterministic trainable assimilation scheme. Both reconstruct complete atmospheric states and operate in an autoregressive fashion with fixed assimilation cycles. In contrast, Appa can generate entire probabilistic trajectories at once, supports both reanalysis and observational forecasting, and enables zero-shot conditioning on any differentiable measurement function without retraining, allowing immediate adaptation to new observation networks or instruments.

Another research direction adapts variational principles with deep learning components. Fengwu-4DVar [49] integrates the Fengwu [41] weather forecasting model with traditional 4D-Var assimilation. 4DVarNet [50, 51] trains neural solvers for variational assimilation of satellite-derived geophysical fields. Though effective in their respective domains, these approaches typically operate at coarser resolutions, produce deterministic point estimates rather than full posterior distributions, and require retraining to accommodate new observation types, limiting their flexibility compared to Appa.

Finally, MetNet3 [52], Aardvark Weather [53] and GraphDOP [40] take a different approach by combining data assimilation with prediction into a single end-to-end process. These models directly generate forecasts from sparse observations without explicitly estimating the full intermediate atmospheric state. While effective for their intended forecasting applications, they lack Appa’s flexibility, particularly due to their tight dependency on specific types of observations at training.

Limitations and future work While Appa shows strong capabilities in global weather data assimilation, several avenues for improvement remain. The most promising direction involves scaling to larger models trained on more historical data. Our current 1.5B-parameter implementation, though effective, likely represents only the beginning of what is possible with this framework. Empirical scaling laws [54, 55] observed in other domains suggest that larger models could significantly improve the representation of rare events and complex atmospheric phenomena.

Another important area for improvement is our current reliance on simplified synthetic observations. Transitioning to more realistic observational data represents a critical next step. Future work should focus on incorporating observation types that better reflect operational settings, such as satellite radiance measurements rather than direct state variables. This would require developing observation

operators that accurately capture both the error characteristics and nonlinearities of diverse measurement instruments. Validating Appa with such realistic observation scenarios would strengthen its case for potential operational deployment.

Related to observation realism is the challenge of physical coherence in generated states. Despite our promising physical consistency checks, there remain substantial opportunities to develop more rigorous diagnostics and improve physical fidelity. The compression of atmospheric states into a latent space inherently sacrifices some small-scale phenomena and information, which may affect physical consistency. Future iterations could explore spatially localized score-based data assimilation approaches that potentially preserve more fine-grained physical features.

Additionally, the blanket mechanism underlying our approach warrants deeper investigation, as we observed that trajectories quickly revert toward the prior distribution when forecasting, potentially explaining the abrupt decline in forecast skill. The information from observations may require multiple diffusion steps to influence temporally distant states, with the risk of signal attenuation over long sequences. Improvements should integrate more sophisticated temporal connectivity structures, alternative blanket sizes and overlap strategies, and better conditioning techniques that balance explicit conditioning (within observed blankets) with implicit conditioning (through diffusion steps) to enhance both physical fidelity and computational efficiency. These improvements may enable all-at-once forecasting to reach performance similar to autoregressive rollouts, while retaining its significant computational advantage.

From a statistical perspective, a key aspect that requires further investigation is the calibration of posterior distributions produced by our framework. While Appa offers access to the full distribution of compatible states, systematic evaluation of these distributions against the ground truth remains to be conducted, particularly in scenarios with sparse or noisy observations. In particular, our posterior sampling algorithm relies on several approximations, including linearization of both the observation process and the decoder, which could guide sampling toward suboptimal regions of the latent space. These approximations may compromise the statistical quality of our generations. Furthermore, it is worth noting that the objectives of Bayesian inference – faithfully representing uncertainty – can sometimes differ from traditional metrics in weather modeling that prioritize point-estimate accuracy. Future work should focus on balancing faithful posterior representation with the practical requirements of weather applications, where strong reconstructions and forecasts are typically prioritized.

While our framework has demonstrated good temporal consistency in assimilation tasks, successfully generating physically coherent trajectories with hourly resolution for extended periods, computational efficiency in the conditioning process remains a challenge. The need to pass through the decoder at every sampling step when conditioning on pixel-space observations creates a substantial computational bottleneck, especially when assimilating large observation sets across extended time windows. Future work could explore projecting observations directly into latent space, which would greatly reduce the overhead of posterior sampling at the cost of learning an additional model to replace our current likelihood approximation. This approach could potentially eliminate the decoder bottleneck during inference while maintaining the probabilistic benefits of our framework.

Finally, we emphasize that Appa represents an initial prototype rather than a finished product. While our results demonstrate promising capabilities, particularly in reanalysis tasks, we see substantial opportunities for enhancement through architectural refinements, improved conditioning mechanisms, and more sophisticated physical consistency constraints. Ongoing work focuses on addressing the limitations identified in this study, with the ultimate goal of developing a system that combines the flexibility of probabilistic generation with the accuracy and efficiency demanded by operational meteorology.

Acknowledgments

Gérôme Andry, François Rozet, Sacha Lewin, and Elise Faulx are research fellows of the *National Fund for Scientific Research* (F.R.S.-FNRS) and acknowledge its financial support. Omer Rochman gratefully acknowledges the financial support of the *Walloon Region* under Grant No. 2010235 (ARIAC by Digital Wallonia 4.AI). Victor Mangeleer is a research fellow part of the *Multiple Threats on Ocean Health* (MITHO) project and gratefully acknowledges funding from the *European Space Agency* (ESA).

Computational resources have been provided by the *Consortium des Équipements de Calcul Intensif* (CÉCI), funded by the *National Fund for Scientific Research* (F.R.S.-FNRS) under Grant No. 2502011 and by the *Walloon Region*, including the Tier-1 supercomputer of the *Wallonia-Brussels Federation*, infrastructure funded by the Walloon Region under Grant No. 1117545

References

- [1] Jaideep Pathak et al. “FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators”. 2022.
- [2] Lei Chen et al. “FuXi: a cascade machine learning forecasting system for 15-day global weather forecast”. In *npj Climate and Atmospheric Science* 6.1 (2023).
- [3] Remi Lam et al. “Learning skillful medium-range global weather forecasting”. In *Science* 382.6677 (2023).
- [4] Ilan Price et al. “Probabilistic weather forecasting with machine learning”. In *Nature* 637.8044 (2025).
- [5] Simon Lang et al. “AIFS – ECMWF’s data-driven forecasting system”. 2024.
- [6] Cristian Bodnar et al. “A Foundation Model for the Earth System”. 2024.
- [7] A. C. Lorenc. “Analysis methods for numerical weather prediction”. In *Quarterly Journal of the Royal Meteorological Society* 112.474 (1986).
- [8] François-Xavier Le Dimet and Olivier Talagrand. “Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects”. In *Tellus A* 38A.2 (1986).
- [9] Yannick Trémolet. “Accounting for an imperfect model in 4D-Var”. In *Quarterly Journal of the Royal Meteorological Society* 132.621 (2006).
- [10] Yannick Trémolet. “Model-error estimation in 4D-Var”. In *Quarterly Journal of the Royal Meteorological Society* 133.626 (2007).
- [11] Mike Fisher et al. “Weak-constraint and long-window 4D-Var”. In *ECMWF Technical Memoranda* 655 (2011).
- [12] Alberto Carrassi et al. “Data assimilation in the geosciences: An overview of methods, issues, and perspectives”. In *WIREs Climate Change* 9.5 (2018).
- [13] François Rozet and Gilles Louppe. “Score-based data assimilation”. In *Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS ’23*. Red Hook, NY, USA: Curran Associates Inc., 2023.
- [14] François Rozet and Gilles Louppe. “Score-based data assimilation for a two-layer quasi-geostrophic model”. In *Machine learning and the physical sciences workshop (NeurIPS)*. 2023.
- [15] Peter Manshausen et al. “Generative Data Assimilation of Sparse Weather Station Observations at Kilometer Scales”. 2025.
- [16] Jonathan Schmidt et al. “A Generative Framework for Probabilistic, Spatiotemporally Coherent Downscaling of Climate Simulation”. 2025.
- [17] Jascha Sohl-Dickstein et al. “Deep unsupervised learning using nonequilibrium thermodynamics”. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. ICML’15*. Lille, France: JMLR.org, 2015.
- [18] Jonathan Ho et al. “Denoising Diffusion Probabilistic Models”. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.
- [19] Yang Song et al. “Score-Based Generative Modeling through Stochastic Differential Equations”. In *International Conference on Learning Representations*. 2020.
- [20] Brian D. O. Anderson. “Reverse-time diffusion equation models”. In *Stochastic Processes and their Applications* 12.3 (1982).
- [21] Simo Särkkä and Arno Solin. “Applied stochastic differential equations”. Vol. 10. Cambridge University Press, 2019.
- [22] Yang Song and Stefano Ermon. “Generative modeling by estimating gradients of the data distribution”. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 1067. Red Hook, NY, USA: Curran Associates Inc., 2019.

- [23] Hyungjin Chung et al. “Diffusion Posterior Sampling for General Noisy Inverse Problems”. In *The Eleventh International Conference on Learning Representations*. 2022.
- [24] François Rozet et al. “Learning Diffusion Priors from Observations by Expectation Maximization”. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024.
- [25] Giannis Daras et al. “A Survey on Diffusion Models for Inverse Problems”. 2024.
- [26] Robin Rombach et al. “High-resolution image synthesis with latent diffusion models”. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2022.
- [27] William Peebles and Saining Xie. “Scalable Diffusion Models with Transformers”. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023.
- [28] Adam Polyak et al. “Movie Gen: A Cast of Media Foundation Models”. 2025.
- [29] Hans Hersbach et al. “The ERA5 global reanalysis”. In *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020).
- [30] Juho Lee et al. “Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks”. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.
- [31] Takeshi D. Itoh et al. “Multi-level attention pooling for graph neural networks: Unifying graph representations with multiple localities”. In *Neural Networks* 145 (2022).
- [32] Maksim Zhdanov et al. “Erwin: A Tree-based Hierarchical Transformer for Large-scale Physical Systems”. 2025.
- [33] Petar Veličković et al. “Graph Attention Networks”. In *International Conference on Learning Representations*. 2018.
- [34] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. 2022.
- [35] Vincent Sitzmann et al. “Implicit Neural Representations with Periodic Activation Functions”. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.
- [36] Marc Rußwurm et al. “Geographic Location Encoding with Spherical Harmonics and Sinusoidal Representation Networks”. In *The Twelfth International Conference on Learning Representations*. 2023.
- [37] Tero Karras et al. “Elucidating the Design Space of Diffusion-Based Generative Models”. In *Advances in Neural Information Processing Systems* 35 (2022).
- [38] Piotr Mirowski et al. “Neural Compression of Atmospheric States”. 2024.
- [39] NOAA National Centers for Environmental Information. “Global surface summary of the day - GSOD”. Version 1.0. 1999.
- [40] Mihai Alexe et al. “GraphDOP: Towards skilful data-driven medium-range weather forecasts learnt and initialised directly from observations”. 2024.
- [41] Kang Chen et al. “FengWu: Pushing the Skillful Global Medium-range Weather Forecast beyond 10 Days Lead”. 2023.
- [42] Aliaksandra Shysheya et al. “On conditional diffusion models for PDE simulations”. In *Advances in Neural Information Processing Systems* 37 (2024).
- [43] F. Rabier et al. “The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics”. In *Quarterly Journal of the Royal Meteorological Society* 126.564 (2000).
- [44] Geir Evensen. “Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics”. In *Journal of Geophysical Research: Oceans* 99.C5 (1994).
- [45] Brian R Hunt et al. “Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter”. In *Physica D: Nonlinear Phenomena* (2007).
- [46] Sibó Cheng et al. “Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review”. In *IEEE/CAA Journal of Automatica Sinica* (2023).
- [47] Langwen Huang et al. “DiffDA: a diffusion model for weather-scale data assimilation”. In *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. ICML’24. Vienna, Austria: JMLR.org, 2024.

- [48] Xiaoze Xu et al. “Fuxi-DA: A Generalized Deep Learning Data Assimilation Framework for Assimilating Satellite Observations”. 2024.
- [49] Yi Xiao et al. “FengWu-4DVar: Coupling the Data-driven Weather Forecasting Model with 4D Variational Assimilation”. 2024.
- [50] Ronan Fablet et al. “Joint interpolation and representation learning for irregularly sampled satellite-derived geophysical fields”. In *Frontiers in Applied Mathematics and Statistics* (2021).
- [51] Ronan Fablet and Bertrand Chapron. “Multimodal learning-based inversion models for the space-time reconstruction of satellite-derived geophysical fields”. 2022.
- [52] Marcin Andrychowicz et al. “Deep Learning for Day Forecasts from Sparse Observations”. 2023.
- [53] Anna Allen et al. “End-to-end data-driven weather prediction”. In *Nature* (2025).
- [54] Jared Kaplan et al. “Scaling Laws for Neural Language Models”. 2020.
- [55] Yasaman Bahri et al. “Explaining neural scaling laws”. In *Proceedings of the National Academy of Sciences* 121.27 (2024).
- [56] European Space Agency (ESA). “Overview of sentinel-3 mission”. 2025.

A Data

A.1 ERA5

ERA5 is a global deterministic reanalysis dataset from ECMWF that provides high-resolution (0.25°) hourly estimates of atmospheric, land, and oceanic variables from 1959 onward [29]. It assimilates observations into a numerical weather prediction model using 4D-Var data assimilation.

For this work, we use a subset of ERA5 data, defined on a 0.25° equiangular grid with 13 pressure levels: 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, and 1000 hPa. Due to storage limitations, we restrict the temporal coverage of the dataset to the 1999–2019 period, with data split into training (1999–2017), validation (2018), and testing (2019).

Table 1 lists the selected variables. Some serve as both input and predicted features, while others provide contextual information (only input). Context variables are not predicted but help define the temporal and dynamic conditions under which predictions are made, improving model performance.

Table 1. Input variables.

Type	Variable Name	Role
Atmospheric	Temperature	Input/Predicted
Atmospheric	U-Wind Component	Input/Predicted
Atmospheric	V-Wind Component	Input/Predicted
Atmospheric	Geopotential	Input/Predicted
Atmospheric	Specific Humidity	Input/Predicted
Single	2m Temperature	Input/Predicted
Single	10m U-Wind Component	Input/Predicted
Single	10m V-Wind Component	Input/Predicted
Single	Mean Sea Level Pressure	Input/Predicted
Single	Sea Surface Temperature	Input/Predicted
Single	Total Precipitation	Input/Predicted
Single	Total Incident Solar Radiation	Input
Clock	Local time of day	Input
Clock	Elapsed year progress	Input

A.2 Data pre-processing

Standardization Although the dynamics across the atmospheric column are correlated, each pressure level exhibits distinct statistical behavior. Thus, we computed the mean and standard deviation separately for each variable and at each pressure level, on the whole training dataset. We used these statistics to standardize our entire dataset and to rescale the output of Appa.

Handling missing values Since Sea Surface Temperature is undefined over land (NaN values), we replace these with zeros as a neutral placeholder after standardization.

Data availability ERA5 data was downloaded from the WeatherBench2 platform, where Google has made them publicly available via Google Cloud Storage.

A.3 Observations

The weather station observations use the physical locations of 11,527 real stations, obtained from NOAA’s NCEI Global Surface Summary of the Day [39]. From these locations, a mask of 10,277 pixels (out of 1,038,240, i.e., $\sim 1\%$) is created.

Satellite observations The satellite observations assume simple circular orbits with no other effects. Given an orbital radius and inclination, a circular orbit is simulated and the satellite’s position is projected onto the Earth. Lastly, all points within a certain radius of the projected points are observed. The temporal aspect of the orbit is dealt with by breaking it into 1h segments, and considering only the observations acquired within that 1h period. The radius was set to 500km, roughly matching the field of view of Sentinel-3 [56].

B Architectures and training details

B.1 Autoencoder

The autoencoder transforms atmospheric data from a high-dimensional rectangular N320 grid to a compact latent representation through progressive downsampling and channel expansion on icosahedral meshes, as detailed in Figure 9 and Table 2.

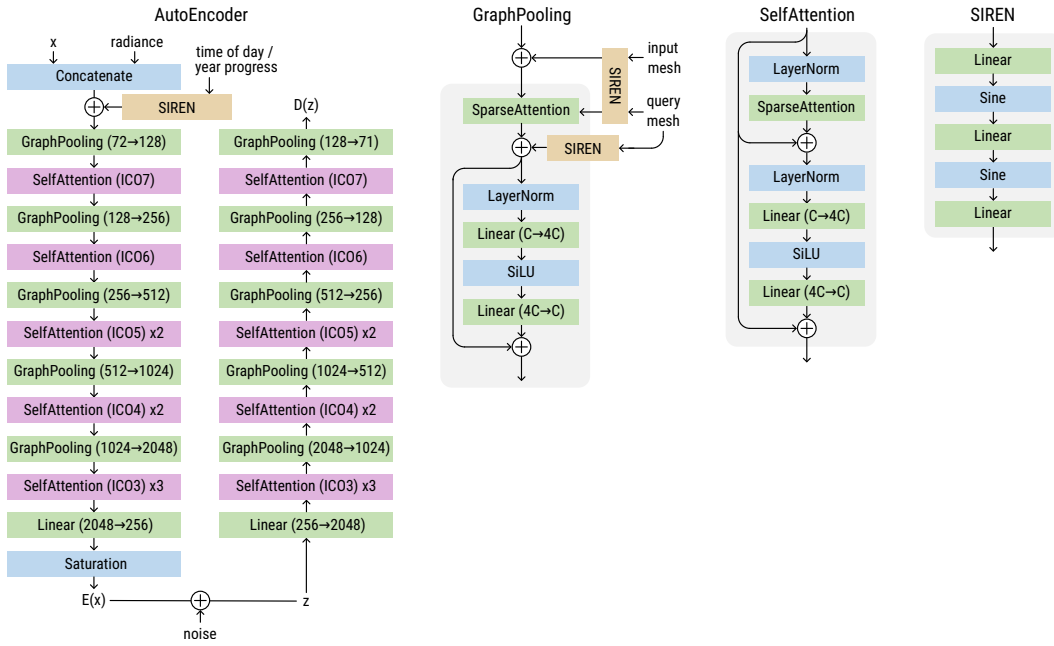


Figure 9. Autoencoder architecture.

Table 2. Mesh hierarchy throughout the autoencoder network.

Mesh	Vertices	Channels
N320	1,038,240	71
ICO7	163,842	128
ICO6	40,962	256
ICO5	10,242	512
ICO4	2,562	1,024
ICO3	642	2,048
Latent	642	256

Table 3. Autoencoder training configuration

Parameter	Value
Loss function	Latitude-weighted mean squared error
Latent noise	$\sigma^2 = 0.1$ for regularization
Optimizer	SOAP with initial learning rate 3×10^{-5} and linear decay
Batch size	64 samples per step
Training duration	45000 update steps (approximately 2 days)
Hardware	64× NVIDIA A100 40GB GPUs

B.2 Latent denoiser

The denoiser operates on latent trajectories, combining spatial and temporal attention to model atmospheric dynamics across both dimensions simultaneously.

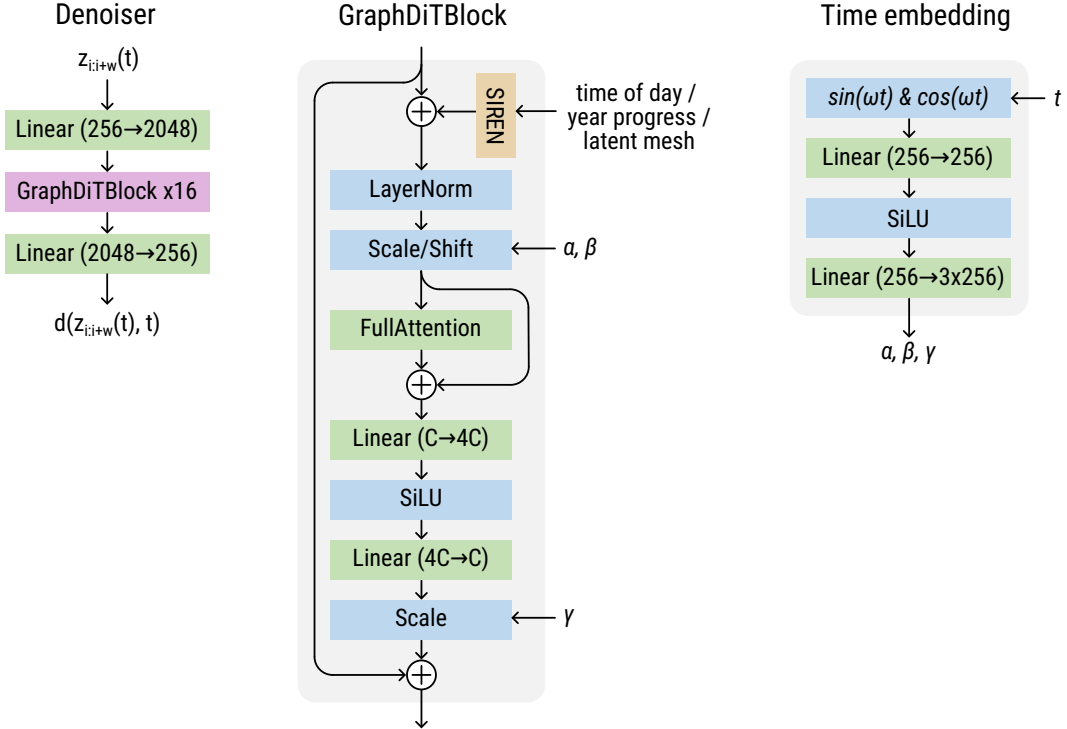


Figure 10. Denoiser architecture.

Table 4. Denoiser training configuration

Parameter	Value
Loss	Denoising score matching with rectified noise schedule
Noise range	$\sigma_{\min} = 0.001, \sigma_{\max} = 1000$
Optimizer	SOAP with initial learning rate 5×10^{-5} and linear decay
Batch size	256 samples per step
Training duration	30000 update steps (approximately 5 days)
Hardware	64× NVIDIA A100 40GB GPUs

C Physical consistency checks

C.1 Altitude

The geopotential Φ is defined as the gravitational potential energy per unit mass at a given altitude H . It can be expressed as an integral of gravitational acceleration from sea level to altitude H . Using Newton's law of universal gravitation, this gives

$$\Phi = \int_0^H g_h \partial h = \int_0^H \frac{GM_e}{(R_e + h)^2} \partial h, \quad (15)$$

where g_h is the gravitational acceleration at altitude h , G is the gravitational constant, and R_e and M_e are the Earth's radius and mass, respectively. Integrating this equation yields

$$\Phi = \frac{GM_e H}{R_e(R_e + H)}. \quad (16)$$

We can express this formula using the Earth's gravitational acceleration at the surface g_0 by substituting

$$G = \frac{g_0 R_e}{M_e}, \quad (17)$$

which gives our first formula for altitude

$$H = \frac{\Phi R_e}{g_0 R_e - \Phi}. \quad (18)$$

In our implementation, we use mean values for g_0 (9.80665 m/s^2) and R_e ($6.378 \times 10^6 \text{ m}$), though these values vary slightly with position on the globe.

For our second approach to calculate altitude, we begin with the hydrostatic equation

$$\frac{\partial p}{\partial h} = -\rho_h g_h, \quad (19)$$

where $\frac{\partial p}{\partial h}$ is the rate of change of pressure with altitude, and ρ_h is the air density (in kg/m^3) at altitude h . From the ideal gas law, we have

$$p_h = \frac{\rho_h R T_h}{M}, \quad (20)$$

where R is the universal gas constant ($8.31462 \text{ J/(mol}\cdot\text{K)}$), M is the approximate atmospheric molar mass (0.02896 kg/mol , assuming 80% nitrogen and 20% oxygen), and P_h and T_h are pressure (Pa) and temperature (K) respectively. Combining these equations

$$\int_{p_0}^{p_H} \frac{1}{p_h} \partial p = \int_0^H \frac{M g_h}{R T_h} \partial h, \quad (21)$$

where p_0 is sea-level pressure and p_H is pressure at altitude H . Assuming R , M , and g_h remain constant (approximating g_h as g_0), we integrate both sides

$$\int_0^H \frac{1}{T_h} \partial h = \frac{R}{M g_0} \ln \left(\frac{p_H}{p_0} \right). \quad (22)$$

To compute altitude from this equation, we use numerical integration via the trapezoidal rule. Each pressure level i corresponds to an altitude

$$h_i = h_{i-1} + \frac{R}{\frac{M g_0}{2} \left(\frac{1}{T_i} + \frac{1}{T_{i-1}} \right)} \ln \left(\frac{p_i}{p_{i-1}} \right). \quad (23)$$

We access all values of p_i and T_i for levels 1-13 directly from our dataset. Starting at sea level ($h_0 = 0$), we use the mean sea level pressure from our surface variables as p_0 . The sea-level temperature T_0 is derived from the potential temperature of the first available temperature using

$$T_0 = T_1 \left(\frac{P_0}{P_1} \right)^{\frac{c_p}{R}}, \quad (24)$$

where c_p is the specific heat capacity of air ($\approx 1005 \text{ J/(kg}\cdot\text{K)}$).

C.2 Wind properties

The pressure difference between two locations (or at a fixed pressure level, the altitude difference) forces the air to move. This is called the pressure gradient force, which equals the gradient of geopotential, a quantity that we can obtain easily from our dataset. As the air gains velocity, the Coriolis force deflects the wind until it flows parallel to the isobars.

If only these two forces are present, an equilibrium establishes between the pressure gradient force and the Coriolis force. Both are perpendicular to the isobars, meaning they no longer alter the wind's direction. This wind is called geostrophic wind,

$$\nabla \Phi = -2 \sin \phi \vec{\Omega} \times \vec{V}_g, \quad (25)$$

where $\nabla \Phi$ corresponds to the pressure gradient force, $\vec{\Omega}$ is a vector parallel to the Earth's rotation axis with a magnitude equal to the Earth's angular velocity (set to its mean value of $7 \cdot 10^{-5} \text{ rad s}^{-1}$ in our implementation), ϕ is the latitude in degrees (ranging from 90° to -90°), and \vec{V}_g is the geostrophic wind.

To compute vector orientations (geopotential and wind), we must account for the Earth's shape. Pixel lengths Δx and Δy along the x (longitude) and y (latitude) directions are defined by

$$\Delta x = \frac{2\pi R_e \cos(\phi)}{N_x}, \quad (26)$$

$$\Delta y = \frac{\pi R_e}{N_y}, \quad (27)$$

where N_x and N_y are the number of pixels along longitude and latitude, respectively. Thus, the partial derivatives of the geopotential can be approximated by

$$\frac{\partial \Phi}{\partial x} = \frac{4\pi \Omega R_e}{N_x} \sin \phi \cos \phi u_g, \quad (28)$$

$$\frac{\partial \Phi}{\partial y} = -\frac{2\pi \Omega R_e}{N_y} \sin \phi v_g. \quad (29)$$

The 500 hPa level is often referred to as the Non-Divergence Level, implying that wind should be mostly geostrophic at this altitude (though some local variations may still occur). Thus, if we replace the geostrophic components u_g and v_g in Equations (28) and (29) with the observed wind components u and v directly available from our dataset, we should obtain a good approximation of the geopotential gradient.

We can evaluate the validity of orientation if the cosine between the gradient and its approximation is close to zero (or equivalently, if the sine is close to -1 or 1). We can also evaluate the validity of magnitude if the Pearson correlation coefficient between the geopotential gradient and its approximation is close to 1.

Near the surface, an ageostrophic wind adds to the geostrophic wind to produce the observed wind. Thus, the gradient approximation (Equations (28) and (29)) is no longer valid if we consider the observed wind components u and v as the geostrophic wind components u_g and v_g . Another formula is needed to compute the ageostrophic wind, which should also include several additional components: centripetal acceleration due to the isobars curvature, which is significant in depression and anticyclonic situations; vertical motion; and friction, which acts against the wind direction and mostly depends on orography, altitude, and atmospheric stability. This leads to the equilibrium equation

$$\nabla \Phi + 2\vec{\Omega} \sin \phi \vec{V}_g + \frac{\vec{V}^2}{R_C} + \vec{w} + \vec{F}_f = \vec{0}, \quad (30)$$

where R_C is the radius of curvature, \vec{w} is the vertical motion, and \vec{F}_f is the friction force.

There should remain an equilibrium between these five forces, leading to a wind that differs from the geostrophic wind in magnitude (correlation lower than 1) and orientation (wind deflected from its parallelism and oriented at 20 – 25° from high to low pressure). In this case, it remains interesting to compute the approximation of the geopotential gradient (Equations (28) and (29)) to analyze if these features are recognized.

D Supplementary results

D.1 Quantitative results

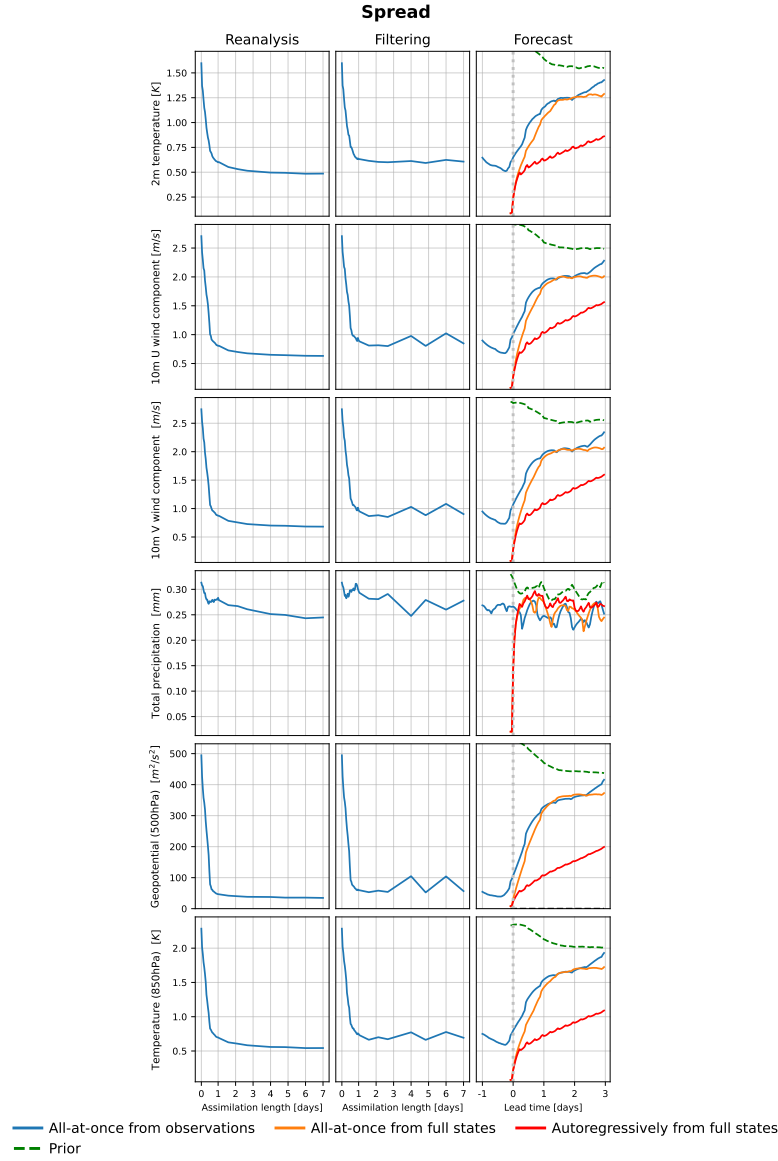


Figure 11. Quantitative evaluation across reanalysis, filtering and forecasting tasks (ensemble spread).

D.2 Qualitative results

Video trajectories and additional qualitative results can be found on the companion website at <https://montefiore-sail.github.io/appa>.