Training-Free Data Assimilation with GenCast

Thomas Savary
ENS Paris-Saclay
thomas.savary@ens-paris-saclay.fr

François Rozet
University of Liège
francois.rozet@uliege.be

Gilles Louppe
University of Liège
g.louppe@uliege.be

Abstract

Data assimilation is widely used in many disciplines such as meteorology, oceanography, and robotics to estimate the state of a dynamical system from noisy observations. In this work, we propose a lightweight and general method to perform data assimilation using diffusion models pre-trained for emulating dynamical systems. Our method builds on particle filters, a class of data assimilation algorithms, and does not require any further training. As a guiding example throughout this work, we illustrate our methodology on GenCast, a diffusion-based model that generates global ensemble weather forecasts.

1 Introduction

Simulating physical phenomena traditionally involves solving partial differential equations with dedicated numerical solvers [1–3]. Recently, neural networks [4] have emerged as a compelling alternative, achieving competitive accuracy at substantially lower computational cost [5–7]. In particular, diffusion models [8, 9] attract growing interest due to their ability to capture high-dimensional, multimodal distributions [10], making them promising candidates for modeling physical systems [11–14]. In this context, the present paper contributes to ongoing efforts to adapt data assimilation algorithms to this new diffusion-based paradigm [15–18].

More precisely, we consider the filtering problem, which aims to estimate the state of a dynamical system at time k from past and present observations $y^{1:k}$, that is, to approximate the posterior distribution $p(x^k \mid y^{1:k})$ [19]. For this purpose, we focus on particle filters [20]. Unlike commonly used data assimilation methods such as 4D-Var [21] or the Ensemble Kalman filter [22], particle filters do not rely on linearizations that may not fully capture highly nonlinear processes, and thus represent a promising alternative. Since GenCast [14], an autoregressive diffusion model for generating global weather predictions, perfectly matches our problem setting, we use it as a case study in what follows.

2 Background

GenCast and diffusion models GenCast [14] is a global, data-driven weather forecasting system based on diffusion models that produces 15-day ensemble forecasts at 12-hour intervals and 0.25° resolution. To achieve this, the globe's surface is discretized into grid points, each described by a set of surface and atmospheric variables (e.g., temperature, humidity, wind components), and the model autoregressively generates probable future states from current states.

More precisly, given x^k the approximation of the complete atmospheric state at time k using surface and atmospheric variables at the different points of the grid, GenCast generates samples from the

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2025.

distribution $p(x^{k+1} \mid x^k)$. To do so, as explained in [14, 23], we solve a stochastic differential equation of the form

$$\mathrm{d}x_t^{k+1} = \left[f_t x_t^{k+1} - \frac{1+\eta^2}{2} g_t^2 \nabla_{x_t^{k+1}} \log p\left(x_t^{k+1} \mid x^k\right) \right] \mathrm{d}t + \eta g_t \mathrm{d}w_t \tag{1}$$

where $\eta \in \mathbb{R}_+$ is a parameter controlling stochasticity, $f_t \in \mathbb{R}$ is the drift coefficient, $g_t \in \mathbb{R}_+$ is the diffusion coefficient, $w_t \in \mathbb{R}^n$ denotes a standard Wiener process and $x_t^{k+1} \in \mathbb{R}^n$ is the sample perturbed with noise at time $t \in [0, 1]$ through a Gaussian kernel $p(x_t^{k+1} \mid x^{k+1}) = \mathcal{N}(\alpha_t x^{k+1}, \sigma_t^2 I)$. The coefficients α_t and σ_t are derived from f_t and g_t such that $p(x_1^{k+1} \mid x^{k+1}) \approx \mathcal{N}(0, \sigma_1^2 I)$ [23].

The score function $\nabla_{x_t^{k+1}} \log p\left(x_t^{k+1} \mid x^k\right)$ in Equation (1) is unknown in practice, but can be approximated by a neural network d_θ called denoiser and trained to minimize the denoising error [24]. The optimal denoiser d_{θ^*} is the mean $\mathbb{E}\left[x^{k+1} \mid x^k, x_t^{k+1}\right]$ of $p(x^{k+1} \mid x^k, x_t^{k+1})$ and is linked to the score function through Tweedie's formula [25, 26] (see Appendix A)

$$\mathbb{E}\left[x^{k+1} \mid x^k, x_t^{k+1}\right] = \alpha_t^{-1} \left[x_t^{k+1} + \sigma_t^2 \nabla_{x_t^k} \log p(x_t^{k+1} \mid x^k) \right] \tag{2}$$

which allows to use $s_\theta(x_t^{k+1}, x^k, t) = \sigma_t^{-2} \left[\alpha_t d_\theta\left(x_t^{k+1}, x^k, t\right) - x_t^{k+1}\right]$ as a score estimate in Equation (1). Thus, drawing noise samples from $p(x_1^{k+1} \mid x^k) \approx \mathcal{N}(0, \sigma_1^2 I)$ and solving Equation (1) from $t = 1$ to 0 with an appropriate discretization scheme [24, 27], we obtain samples from $p(x^{k+1} \mid x^k)$.

Particle filters The aim of particle filters [20] is to approximate the filtering posterior distribution $p(x^k \mid y^{1:k})$ by a finite discrete probability measure $\mu_x^k = \sum_{i=1}^N w_i^k \delta_{x_i^k}$ such that the following converges weakly

$$\sum_{i=1}^N w_i^k g(x_i^k) \underset{N \to +\infty}{\longrightarrow} \int g(x^k) p(x^k \mid y^{1:k}) \mathrm{d}x^k \tag{3}$$

where x_i^k are the particles at time step k, w_i^k the associated weights, $y^{1:k}$ the observations and g a continuous and bounded function. In their standard form, particle filters alternate between a sampling step, where particles are propagated using a proposal distribution, and a weighting step, where weights are updated according to the proposal.

In this work, we focus on the "optimal" proposal $p(x^{k+1} \mid x^k, y^{k+1})$ [28], which propagates particles from step k to $k+1$ conditionally on the next observation y^{k+1}. This proposal is coined "optimal" as it minimizes the variance of the weights, which can then be computed recursively as

$$w_i^{k+1} := p(y^{k+1} \mid x_i^k) \times w_i^k. \tag{4}$$

The main drawback of particle filters is the degeneracy of the algorithm, which corresponds to the situation where only a small subset of particles have non-negligible weights. This is due to the dimension of the observation space: the higher this dimension, the more peaked the likelihood is, and the more unlikely it is for the majority of particles to end up close to all the observations [29].

3 Methodology

Sampling from the optimal proposal distribution The use of the optimal proposal distribution suggests that we are able to draw samples from $p(x^{k+1} \mid x^k, y^{k+1})$, which is not often the case in practice. However, for diffusion models like GenCast, this can be done relatively easily by using the posterior score $\nabla_{x_t^{k+1}} \log p\left(x_t^{k+1} \mid x^k, y^{k+1}\right)$ when solving Equation (1) [17, 23].

Thanks to Bayes' rule, the posterior score $\nabla_{x_t^{k+1}} \log p\left(x_t^{k+1} \mid x^k, y^{k+1}\right)$ can be decomposed into two terms as [23, 30]

$$\nabla_{x_t^{k+1}} \log p\left(x_t^{k+1} \mid x^k, y^{k+1}\right) = \nabla_{x_t^{k+1}} \log p\left(x_t^{k+1} \mid x^k\right) + \nabla_{x_t^{k+1}} \log p\left(y^{k+1} \mid x_t^{k+1}, x^k\right). \tag{5}$$

As an estimate of the first term is already available via the pre-trained denoiser (see Section 2), the remaining task is to estimate the likelihood score $\nabla_{x_t^{k+1}} \log p\left(y^{k+1} \mid x_t^{k+1}, x^k\right)$. To do so, assuming

a differentiable observation operator $\mathcal{H}$, a diagonal covariance matrix Σ_y for the observations and a Gaussian forward process $p(y^{k+1} \mid x^{k+1}) = \mathcal{N}(y^{k+1} \mid \mathcal{H}(x^{k+1}), \Sigma_y)$, we evaluate the likehood score as [31]

$$\nabla_{x_t^{k+1}} \log p(y^{k+1} \mid x_t^{k+1}, x^k) = \nabla_{x_t^{k+1}} \mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k]^T H^T \left(\Sigma_y + HVH^T\right)^{-1} v^{k+1} \tag{6}$$

where H is the Jacobian of $\mathcal{H}$, $V = \mathbb{V}[x^{k+1} \mid x_t^{k+1}, x^k]$ and $v^{k+1} = y^{k+1} - \mathcal{H}(\mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k])$. Despite its complex form, this term can be computed efficiently via automatic differentiation and using a linear solver (see [31] or Appendix B for more details).

Computing weights Updating the weights in the case of the optimal proposal is non-trivial because we cannot evaluate $p(y^{k+1} \mid x^k)$ directly in Equation (4). We then propose to approximate $p(x^{k+1} \mid x^k)$ by a Dirac distribution [32] at $\mathbb{E}[x^{k+1} \mid x^k]$ so that

$$p(y^{k+1} \mid x^k) = \int p(y^{k+1} \mid x^{k+1}) p(x^{k+1} \mid x^k) \mathrm{d}x^{k+1} \approx p(y^{k+1} \mid \mathbb{E}[x^{k+1} \mid x^k]). \tag{7}$$

The conditional expectation $\mathbb{E}[x^{k+1} \mid x^k]$ is not known a priori, but can be efficiently estimated using the pre-trained denoiser

$$\mathbb{E}[x^{k+1} \mid x^k] \underset{\varepsilon \sim \mathcal{N}(0,I)}{=} \mathbb{E}[x^{k+1} \mid x^k, \sigma_1 \varepsilon] \approx d_\theta \left(x_{t=1}^{k+1} = \sigma_1 \varepsilon, x^k, t = 1\right) \tag{8}$$

These two elements enable the use of the Fully-Adapted Auxiliary Particle Filter (FA-APF), a particle filter algorithm adapted to the optimal proposal [33] and described in Algorithm 1. We introduce an inflation coefficient α to control the degeneracy of the weights, at the expense of a bias in the approximation of the posterior distribution $p(x^k \mid y^{1:k})$.

Algorithm 1 Fully-Adapted Auxiliary Particle Filter (FA-APF)

Require: initial condition x^0, number of particles N, thresholds $N_{\text{thr}}^{\text{min,max}}$, number of steps K.

$x_i^0 \leftarrow x^0$
$w_i^0 \leftarrow 1/N$
for k in $0, ..., K-1$ **do**
  $\mu_i^{k+1} \leftarrow \mathbb{E}[x^{k+1} \mid x_i^k]$
  while N_{eff} not in $[N_{\text{thr}}^{\text{min}}, N_{\text{thr}}^{\text{max}}]$ **do**
    update/initialize α
    $\hat{w}_i^{k+1} \leftarrow \left[p(y^{k+1} \mid \mu_i^{k+1})\right]^\alpha$
    $w_i^{k+1} \leftarrow \hat{w}_i^{k+1} / \sum_{j=1}^N \hat{w}_j^{k+1}$
    $N_{\text{eff}} \leftarrow 1 / \sum_{i=1}^N (w_i^{k+1})^2$
  $a_i^{k+1} \sim \text{Cat}(\{w_i^{k+1}\}_{1 \le i \le N})$
  $x_i^{k+1} \leftarrow p(x^{k+1} \mid x_{a_i^{k+1}}^k, y^{k+1})$
return $\mu_x^k = \frac{1}{N} \sum_{i=1}^N \delta_{x_i^k}$ for all $k \in [1, K]$

4 Results

Experimental setup To evaluate our method, we apply Algorithm 1 with the pre-trained GenCast denoiser at 1° resolution and $N = 256$ particles ($N_{\text{thr}}^{\min} = 60$, $N_{\text{thr}}^{\max} = 70$). The observations y^k and the initial condition x^0 are obtained from a reference ERA5 trajectory (a global atmospheric reanalysis produced by ECMWF [34]). We only observe temperature at the surface and at all pressure levels on a regular latitude–longitude grid, taking one point every four degrees in both directions, with zero-mean Gaussian noise and a standard deviation of 0.1 Kelvin. This setup corresponds to a linear observation operator $\mathcal{H}$ and a covariance matrix $\Sigma_y = (0.1)^2 I$ for the observations. Following [13], we solve Equation (1) using a temporal discretization of 40 time steps and a third-order Adam-Bashforth scheme [35] with two correction steps [23] and two BiCGStab [36] iterations to solve the linear system of Equation (6).

First, to validate the correctness of the sampling from the optimal proposal (see Section 3), we verify the consistency of the observations y^{k+1} with the conditional posterior predictive distribution $q(\tilde{y}^{k+1} \mid x_i^k, y^{k+1}) = \mathbb{E}_{q(x^{k+1}|x_i^k, y^{k+1})}\left[p(\tilde{y}^{k+1} \mid x^{k+1})\right]$. An example for a specific variable (surface temperature) at an arbitrarily chosen point of the grid is shown in Figure 1.

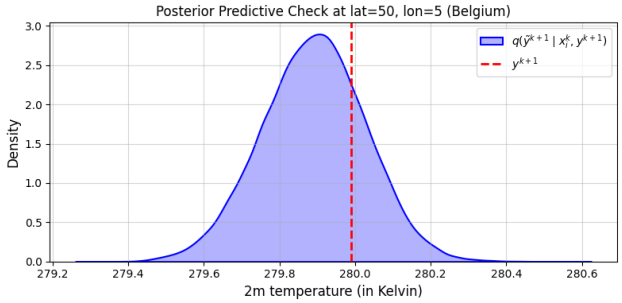

Figure 1: Conditional posterior predictive distribution (blue curve) and corresponding observation (red dashed line) at an arbitrarily chosen point of the grid with coordinates (lat=50, lon=5) for the surface temperature variable. The observation is consistent with the posterior predictive distribution.

Then, we compare the skill (RMSE of the ensemble mean, lower the better) at each time step for the FA-APF and an ensemble of $N = 256$ unconditional forecasts generated autoregressively using Equation (1) without conditioning the score on observations. The skill is computed using the reference ERA5 trajectory from which observations are extracted as ground truth. Figure 2 shows that the skill of the filter reaches a plateau for all variables (including unobserved ones), which is well below the skill of unconditional trajectories. Further results, including skill scores for additional variables, ensemble spread, and trajectory visualizations are presented in Appendix C.

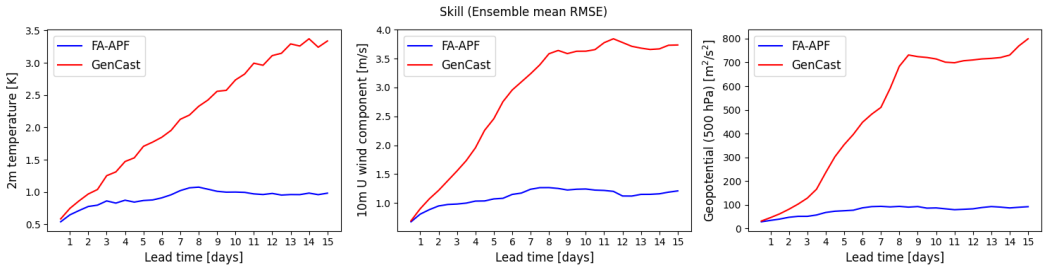

Figure 2: Skill comparison between the FA-APF (blue curve) and the ensemble of unconditional GenCast trajectories (red curve) for the surface temperature (left), the surface U component of wind (middle) and the geopotential at 500 hPA (right). The FA-APF allows to obtain a low and more or less constant skill after 7 days of observations, even for unobserved variables.

5 Conclusion

This work introduces a training-free data assimilation method that shows promising results with GenCast and could be deployed in operational settings with minimal effort. Since it requires no additional training, the approach is readily applicable to other autoregressive diffusion models, and extending its evaluation beyond GenCast is a natural next step [13, 37, 38].

Future work will investigate the role of the initial condition x^0, the thresholds $N_{\text{thr}}^{\text{min,max}}$, and the number of particles N used as hyperparameters in Algorithm 1. Another research direction is to extend this work toward a training-free approach to the more complex problem of reanalysis, which seeks to estimate the state of a dynamical system from past, present, and future observations [19].

Acknowledgments and Disclosure of Funding

François Rozet is a research fellow of the F.R.S.-FNRS (Belgium) and acknowledges its financial support.

We gratefully acknowledge the support of NVIDIA for supporting this research project through the NVIDIA Academic Grant Program.

References

[1] AKIO ARAKAWA and VIVIAN R. LAMB. Computational design of the basic dynamical processes of the ucla general circulation model. In JULIUS CHANG, editor, *General Circulation Models of the Atmosphere*, volume 17 of *Methods in Computational Physics: Advances in Research and Applications*, pages 173–265. Elsevier, 1977. doi: https://doi.org/10.1016/B978-0-12-460817-7.50009-4. URL https://www.sciencedirect.com/science/article/pii/B9780124608177500094.

[2] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of Computation*, 22(104):745–762, 1968. ISSN 00255718, 10886842. URL http://www.jstor.org/stable/2004575.

[3] Charles K. Birdsall and A. Bruce Langdon. *Plasma physics via computer simulation*. Taylor and Francis, New York, 2005. ISBN 0750310251 9780750310253.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[5] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating Eulerian fluid simulation with convolutional networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3424–3433. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/tompson17a.html.

[6] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.

[7] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. doi: 10.1126/science.adi2336. URL https://www.science.org/doi/abs/10.1126/science.adi2336.

[8] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

[11] Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard E. Turner, and Emile Mathieu. On conditional diffusion models for

pde simulations. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

[12] Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 45259–45287. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8df90a1440ce782d1f5607b7a38f2531-Paper-Conference.pdf.

[13] François Rozet, Ruben Ohana, Michael McCabe, Gilles Louppe, François Lanusse, and Shirley Ho. Lost in latent space: An empirical study of latent diffusion models for physics emulation, 2025. URL https://arxiv.org/abs/2507.02608.

[14] I. Price, A. Sanchez-Gonzalez, F. Alet, T. R. Andersson, A. El-Kadi, D. Masters, T. Ewalds, J. Stott, S. Mohamed, P. Battaglia, R. Lam, and M. Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, jan 2025. doi: 10.1038/s41586-024-08252-9. Epub 2024 Dec 4; PMID: 39633054; PMCID: PMC11666454.

[15] Langwen Huang, Lukas Gianinazzi, Yuejiang Yu, Peter D. Dueben, and Torsten Hoefler. Diffda: a diffusion model for weather-scale data assimilation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[16] Jing-An Sun, Hang Fan, Junchao Gong, Ben Fei, Kun Chen, Fenghua Ling, Wenlong Zhang, Wanghan Xu, Li Yan, Pierre Gentine, and Lei Bai. Align-da: Align score-based atmospheric data assimilation with multiple preferences, 2025. URL https://arxiv.org/abs/2505.22008.

[17] Gérôme Andry, François Rozet, Sacha Lewin, Omer Rochman, Victor Mangeleer, Matthias Pirlet, Elise Faulx, Marilaure Grégoire, and Gilles Louppe. Appa: Bending weather dynamics with latent diffusion models for global data assimilation, 2025. URL https://arxiv.org/abs/2504.18720.

[18] Feng Bao, Hristo G. Chipilski, Siming Liang, Guannan Zhang, and Jeffrey S. Whitaker. Nonlinear ensemble filtering with diffusion models: Application to the surface quasi-geostrophic dynamics. *Monthly Weather Review*, 153(7):1155 – 1169, 2025. doi: 10.1175/MWR-D-24-0069.1. URL https://journals.ametsoc.org/view/journals/mwre/153/7/MWR-D-24-0069.1.xml.

[19] 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.

[20] Peter Jan van Leeuwen, Hans R. Künsch, Lars Nerger, Roland Potthast, and Sebastian Reich. Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*, 145(723):2335–2365, 2019. doi: https://doi.org/10.1002/qj.3551. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3551.

[21] 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. doi: 10.3402/tellusa.v38i2.11706. URL https://doi.org/10.3402/tellusa.v38i2.11706.

[22] Geir Evensen. *Data Assimilation : The Ensemble Kalman Filter*. Springer, Berlin, second edition, 2009. ISBN 978-3-642-03710-8. doi: 10.1007/978-3-642-03711-5. URL https://link.springer.com/book/10.1007/978-3-642-03711-5.

[23] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

[24] Tero Karras, Miika Aittala, Samuli Laine, and Timo Aila. Elucidating the design space of diffusion-based generative models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[25] M. C. K. Tweedie. Functions of a statistical variate with given means, with special reference to laplacian distributions. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43 (1):41–49, 1947. doi: 10.1017/S0305004100023185.

[26] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. ISSN 01621459. URL http://www.jstor.org/stable/23239562.

[27] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023. URL https://arxiv.org/abs/2211.01095.

[28] Chris Snyder, Thomas Bengtsson, and Mathias Morzfeld. Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, 143(11):4750 – 4761, 2015. doi: 10.1175/MWR-D-15-0144.1. URL https://journals.ametsoc.org/view/journals/mwre/143/11/mwr-d-15-0144.1.xml.

[29] Peter Van Leeuwen. Aspects of particle filtering in high-dimensional spaces. In Sai Ravela and Adrian Sandu, editors, *Dynamic data-driven environmental systems science*, volume 8964 of *Lecture notes in computer science*, pages 251–262. Springer, Heidelberg, 2015. ISBN 9783319251370. doi: 10.1007/978-3-319-25138-7. URL https://centaur.reading.ac.uk/50238/.

[30] Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems, 2024. URL https://arxiv.org/abs/2209.14687.

[31] François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

[32] Patrick Billingsley. *Probability and measure*. A Wiley-Interscience publication. Wiley, New York [u.a.], 3. ed edition, 1995. ISBN 0471007102. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+164761632&sourceid=fbw_bibsonomy.

[33] François Desbouvries, Yohan Petetin, and Emmanuel Monfrini. Optimal sir algorithm vs. fully adapted auxiliary particle filter: A matter of conditional independence. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3992–3995, 2011. doi: 10.1109/ICASSP.2011.5947227.

[34] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: https://doi.org/10.1002/qj.3803. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803.

[35] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I Nonstiff problems*. Springer, Berlin, second edition, 2000.

[36] H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2): 631–644, 1992. doi: 10.1137/0913035. URL https://doi.org/10.1137/0913035.

[37] Erik Larsson, Joel Oskarsson, Tomas Landelius, and Fredrik Lindsten. Diffusion-lam: Probabilistic limited area weather forecasting with diffusion. In *ICLR 2025 Workshop on Tackling Climate Change with Machine Learning*, 2025. URL https://www.climatechange.ai/papers/iclr2025/36.

[38] Tobias Sebastian Finn, Charlotte Durand, Alban Farchi, Marc Bocquet, Pierre Rampal, and Alberto Carrassi. Generative diffusion for regional surrogate models from sea-ice simulations. *Journal of Advances in Modeling Earth Systems*, 16(10):e2024MS004395, 2024. doi: https://doi.org/10.1029/2024MS004395. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2024MS004395. e2024MS004395 2024MS004395.

A Tweedie's formulae

Theorem A.1. *Assuming that $p(x_t^k \mid x^k) = \mathcal{N}(x_t^k \mid \alpha_t x^k, \sigma_t^2 I)$ and that x_t^{k+1} is conditionally independent of x^k given x^{k+1}, the first moment of the distribution $p(x_t^{k+1} \mid x^k)$ is linked to the score function $\nabla_{x_t^{k+1}} \log p(x_t^{k+1} \mid x^k)$ used in Equation* (1) *through*

$$\nabla_{x_t^{k+1}} \log p(x_t^{k+1} \mid x^k) = \sigma_t^{-2} \left(\alpha_t \mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k] - x_t^{k+1}\right) \tag{9}$$

We provide proofs of this theorem for completeness, even though it is a well known result [25, 26].

Proof.

$$\begin{aligned}
\nabla_{x_t^{k+1}} \log p(x_t^{k+1} \mid x^k) &= \frac{1}{p(x_t^{k+1} \mid x^k)} \nabla_{x_t^{k+1}} p(x_t^{k+1} \mid x^k) \\
&= \frac{1}{p(x_t^{k+1} \mid x^k)} \int \nabla_{x_t^{k+1}} p(x_t^{k+1}, x^{k+1} \mid x^k) \mathrm{d}x^{k+1} \\
&= \frac{1}{p(x_t^{k+1} \mid x^k)} \int p(x_t^{k+1}, x^{k+1} \mid x^k) \nabla_{x_t^{k+1}} \log p(x_t^{k+1}, x^{k+1} \mid x^k) \mathrm{d}x^{k+1} \\
&= \int p(x^{k+1} \mid x_t^{k+1}, x^k) \nabla_{x_t^{k+1}} \log p(x_t^{k+1} \mid x^{k+1}) \mathrm{d}x^{k+1} \\
&= \int p(x^{k+1} \mid x_t^{k+1}, x^k) \sigma_t^{-2} \left(\alpha_t x^{k+1} - x_t^{k+1}\right) \mathrm{d}x^{k+1} \\
&= \alpha_t \sigma_t^{-2} \int x^{k+1} p(x^{k+1} \mid x_t^{k+1}, x^k) \mathrm{d}x^{k+1} - \sigma_t^{-2} x_t^{k+1} \int p(x^{k+1} \mid x_t^{k+1}, x^k) \mathrm{d}x^{k+1} \\
&= \alpha_t \sigma_t^{-2} \mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k] - \sigma_t^{-2} x_t^{k+1} \\
&= \sigma_t^{-2} \left(\alpha_t \mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k] - x_t^{k+1}\right)
\end{aligned}$$

□

B Moment Matching Posterior Sampling

We provide technical details on how $\nabla_{x_t^{k+1}} \log p(y^{k+1} \mid x_t^{k+1}, x^k)$ is estimated for completeness, even though it is already explained in [31].

In order to generate samples conditionally on y^{k+1}, we need to evaluate $\nabla_{x_t^{k+1}} \log p(y^{k+1} \mid x_t^{k+1}, x^k)$ and plug it into equation (5). To do so, we can first write $p(y^{k+1} \mid x_t^{k+1}, x^k)$ as an integral

$$p(y^{k+1} \mid x_t^{k+1}, x^k) = \int p(y^{k+1}, x^{k+1} \mid x_t^{k+1}, x^k) \mathrm{d}x^{k+1} \tag{10}$$

$$= \int p(y^{k+1} \mid x^{k+1}) p(x^{k+1} \mid x_t^{k+1}, x^k) \mathrm{d}x^{k+1} \tag{11}$$

Then, assuming a differentiable observation operator $\mathcal{H}$, a diagonal covariance matrix Σ_y for the observations, a Gaussian forward process $p(y^{k+1} \mid x^{k+1}) = \mathcal{N}(y^{k+1} \mid \mathcal{H}(x^{k+1}), \Sigma_y)$ and a Gaussian approximation $q(x^{k+1} \mid x_t^{k+1}, x^k) = \mathcal{N}(x^{k+1} \mid \mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k], \mathbb{V}[x^{k+1} \mid x_t^{k+1}, x^k])$ of $p(x^{k+1} \mid x_t^{k+1}, x^k)$, we obtain the following approximation

$$q(y^{k+1} \mid x_t^{k+1}, x^k) = \int p(y^{k+1} \mid x^{k+1}) q(x^{k+1} \mid x_t^{k+1}, x^k) \mathrm{d}x^{k+1} \tag{12}$$

$$= \mathcal{N}\left(y^{k+1} \mid \mathcal{H}(\mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k]), \Sigma_y + H \mathbb{V}[x^{k+1} \mid x_t^{k+1}, x^k] H^T\right) \tag{13}$$

where H is the Jacobian of $\mathcal{H}$. This approximation allows to estimate $\nabla_{x_t^{k+1}} \log p(y^{k+1} \mid x_t^{k+1}, x^k)$, under the assumption that the derivative of $\mathbb{V}[x^{k+1} \mid x_t^{k+1}, x^k]$ with respect to x_t^{k+1} is negligible, as

$$\nabla_{x_t^{k+1}} \log q(y^{k+1} \mid x_t^{k+1}, x^k) = \nabla_{x_t^{k+1}} \mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k]^T H^T \left(\Sigma_y + HVH^T\right)^{-1} v^{k+1} \quad (14)$$

where $v^{k+1} = y^{k+1} - \mathcal{H}(\mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k])$ and $V = \mathbb{V}[x^{k+1} \mid x_t^{k+1}, x^k]$. Although Equation (14) gives an explicit formula to estimate $\nabla_{x_t^{k+1}} \log p(y^{k+1} \mid x_t^{k+1}, x^k)$, solving it in practice is not trivial. Indeed, if the dimension of system state is large, compute and store $\mathbb{V}[x^{k+1} \mid x_t^{k+1}, x^k]$ is impossible. However, as $\Sigma_y + H\mathbb{V}[x^{k+1} \mid x_t^{k+1}, x^k]H^T$ is symmetric positive definite (SPD), we can apply the conjugate gradient method. This method is an iterative algorithm to solve linear systems of form $Mv = b$ (where M is SPD), using only implicit access to M through a matrix-vector operator. In our case, the linear system to solve is

$$v^{k+1} = \left(\Sigma_y + H\mathbb{V}[x^{k+1} \mid x_t^{k+1}, x^k]H^T\right) v \quad (15)$$

$$= \Sigma_y v + \alpha_t^{-1}\sigma_t^2 H(\underbrace{v^T H \nabla_{x_t^{k+1}} \mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k]}_{\text{vector-Jacobian product}})^T \quad (16)$$

Within automatic differentiation frameworks, the vector-Jacobian product on the right-hand side can be cheaply evaluate using the pre-trained denoiser as an estimator of $\mathbb{E}[x^{k+1} \mid x_t^{k+1}, x^k]$.

C Supplementary results

C.1 Skill

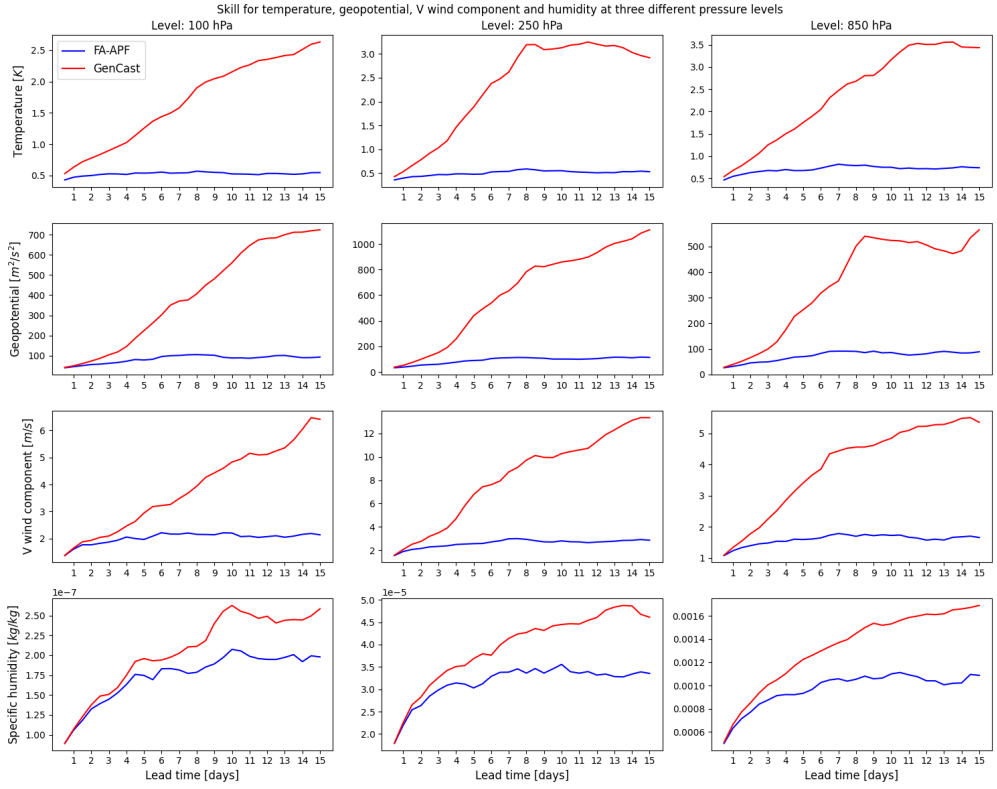

Figure 3: Skill for temperature, geopotential, V component of wind and specific humidity at three different pressure levels (100, 250 and 850 hPa). The skill reaches a plateau after a certain number of time steps for all variables (even those that are not observed), well below the one of GenCast's forecasts.

C.2 Spread

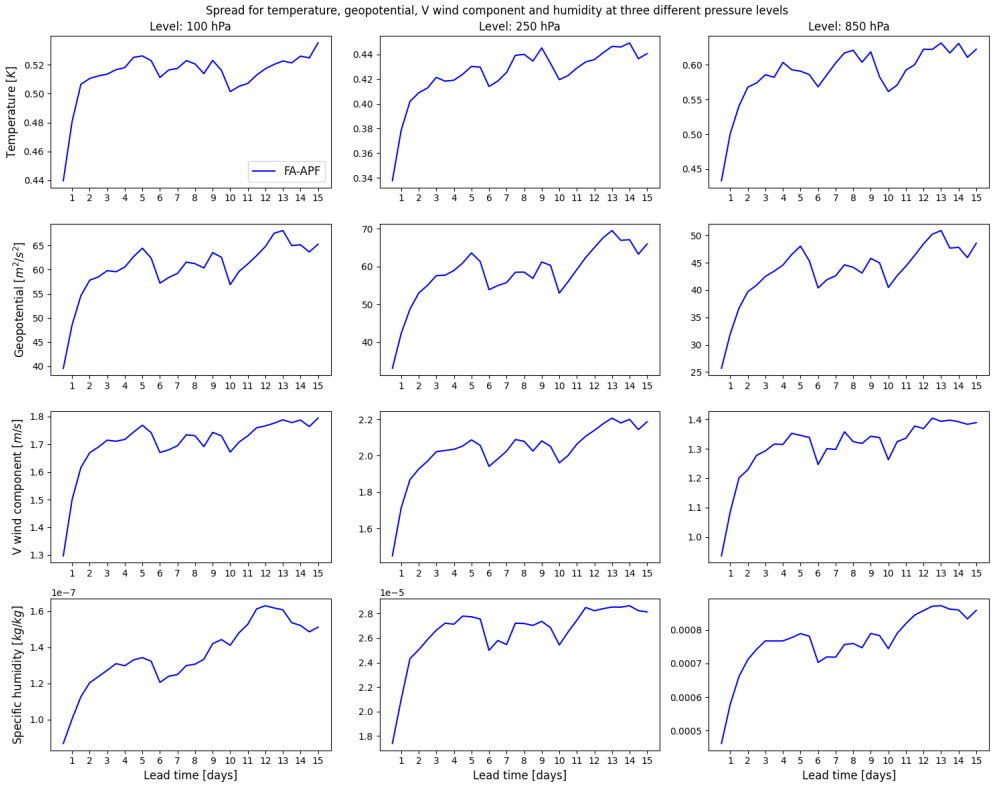

Figure 4: Spread for temperature, geopotential, V component of wind and specific humidity at three different pressure levels (100, 250 and 850 hPa). The spread is non-zero and of the same order of magnitude as the skill, indicating that we capture a distribution rather than collapsing onto a single mode.

C.3 Visualization of trajectories

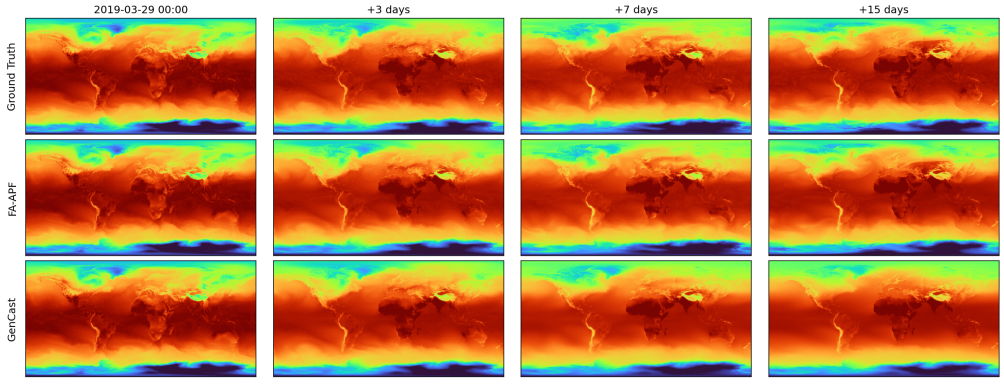

Figure 5: Comparison of surface temperature between the reference ERA5 trajectory (first row), the FA-APF ensemble mean (second row), and the GenCast ensemble mean (third row) after 3, 7, and 15 days.

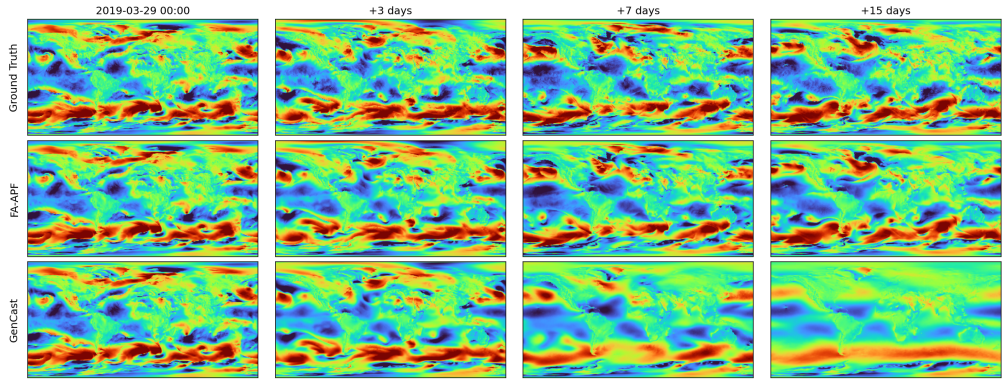

Figure 6: Comparison of the 10m U component of wind between the reference ERA5 trajectory (first row), the FA-APF ensemble mean (second row), and the GenCast ensemble mean (third row) after 3, 7, and 15 days.

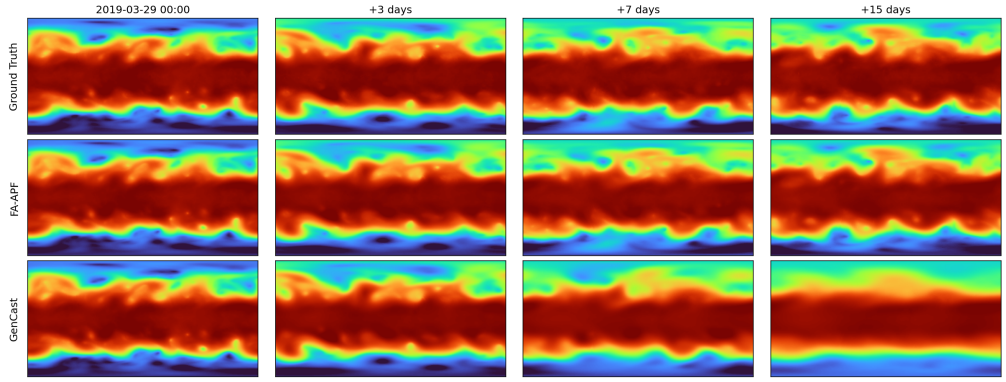

Figure 7: Comparison of the geopotential at 500 hPA between the reference ERA5 trajectory (first row), the FA-APF ensemble mean (second row), and the GenCast ensemble mean (third row) after 3, 7, and 15 days.