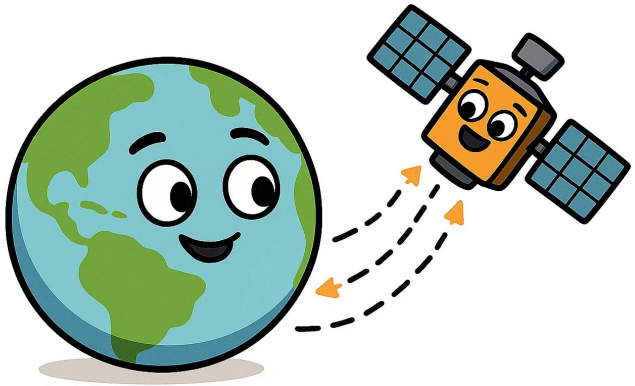
Training-Free Data Assimilation with GenCast

Climat Change AI workshop, NeurIPS 2025
December 7th 2025



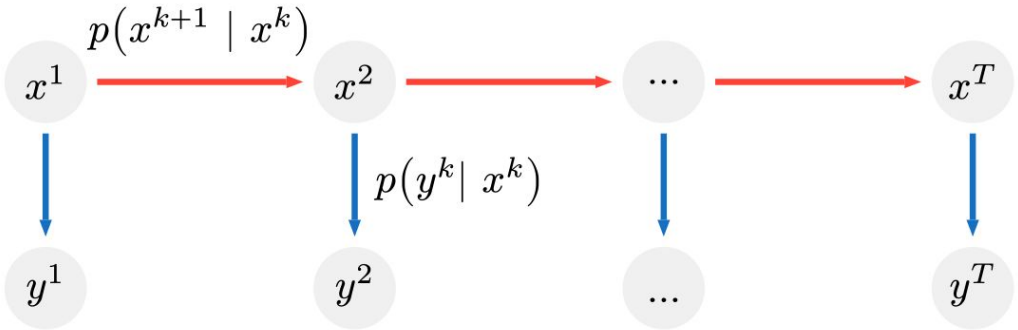
Thomas Savary, François Rozet and Gilles Louppe (University of Liège)

Problem statement

Data assimilation: introduction

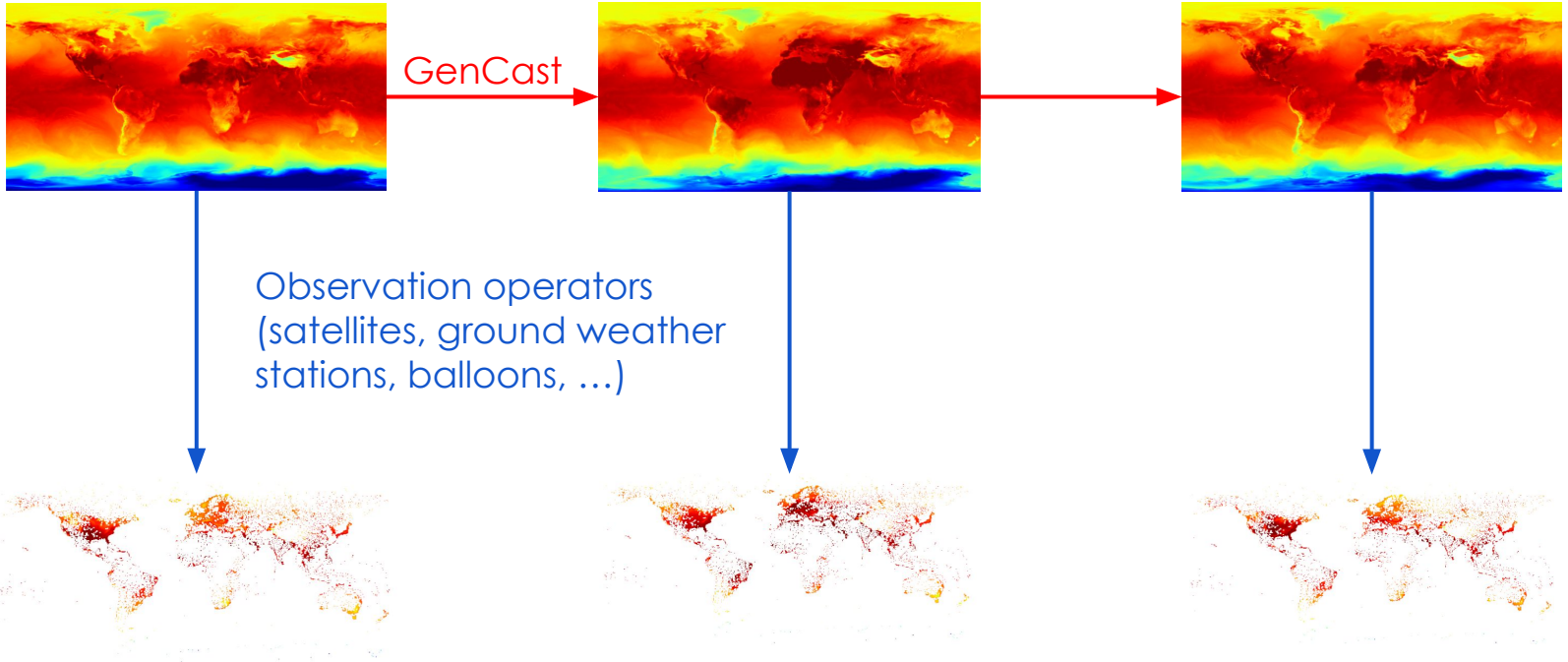
- Data assimilation (DA) algorithms address the problem of inferring the state of a discrete time Markovian dynamical system by using prior knowledge as well as observations.

- **Filtering** is a subproblem of DA that aims to estimate the system state using only current and past observations, i.e., to approximate the posterior distribution $p(x^k \mid y^{1:k})$.



- In this work, the prior knowledge is modeled by a diffusion-based emulator (like GenCast).

Data assimilation: the example of atmosphere



Method & contributions

Particle filters

- Approximate the posterior at each time step by a discrete probability measure

$$p(x^k \mid y^{1:k}) \approx \sum_{i=1}^{N} w_i^k \delta_{x_i^k}$$

- **Sampling step**: current particles are propagated to the next time step via a proposal distribution $q(x^{k+1} \mid x^k, y^{k+1})$.

- **Weighting step**: weights are then updated according to a formula that depends on the proposal distribution.

They can handle highly non-linear dynamics without any assumptions!

But suffer from the curse of dimensionality.

Sampling from the optimal proposal

The use of the optimal proposal distribution suggests that we are able to draw samples from $p(x^{k+1} \mid x^k, y^{k+1})$. This can be done relatively easily for diffusion models by using the **posterior score** $\nabla_{x_t^{k+1}} \log p\left(x_t^{k+1} \mid x^k, y^{k+1}\right)$ during the reverse diffusion process

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

Thanks to Bayes' rule, the posterior score can be decomposed into two terms as

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

The first is already known via the pretrained GenCast denoiser whereas the second one is computed following Rozet et al, 2024.

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. We therefore propose the following approximation

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

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

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

In practice, to control the degeneracy of the algorithm we introduce an inflation coefficient when computing the weights at the expense of a bias.

Results

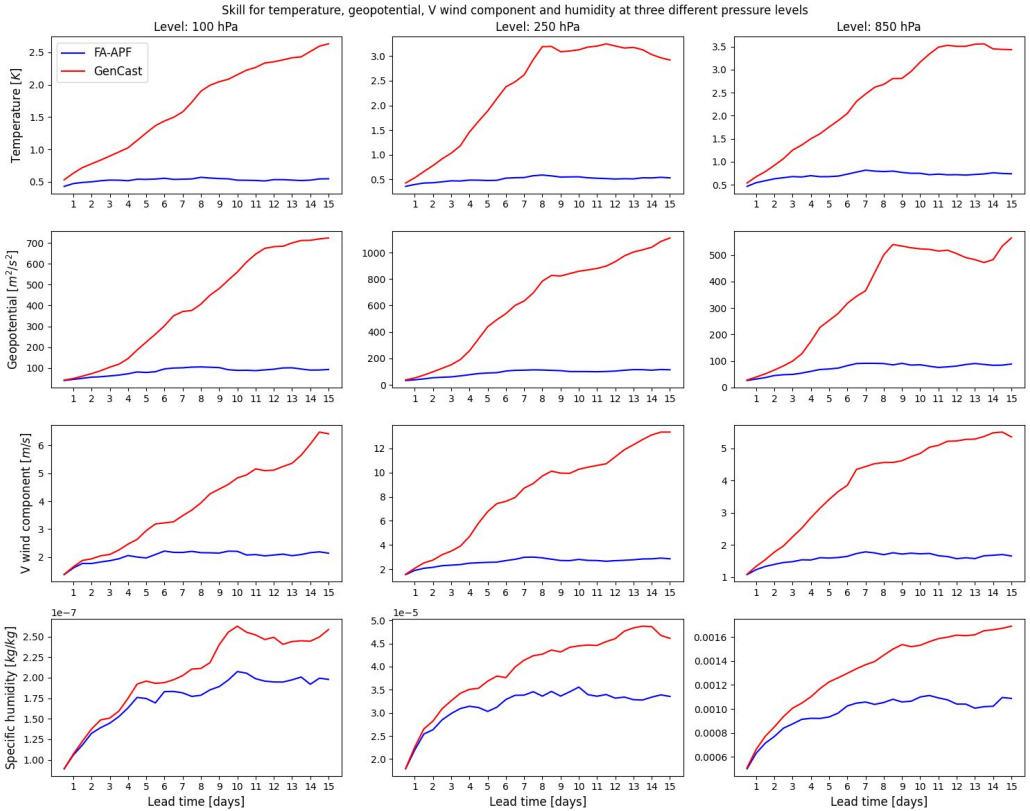


Figure 1. The skill reaches a plateau after a certain number of time steps for all variables (even those that are not observed).

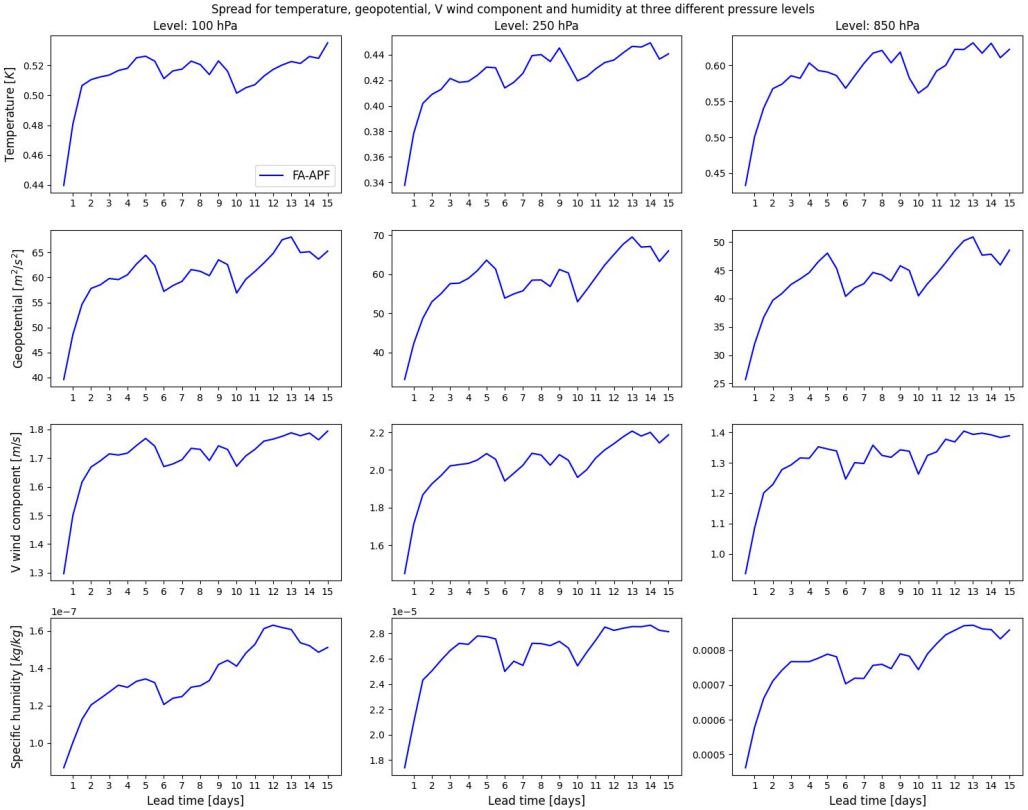


Figure 2. The spread is non-zero and of the same order of magnitude as the skill, indicating that we capture a distribution rather than collapsing onto a single mode.

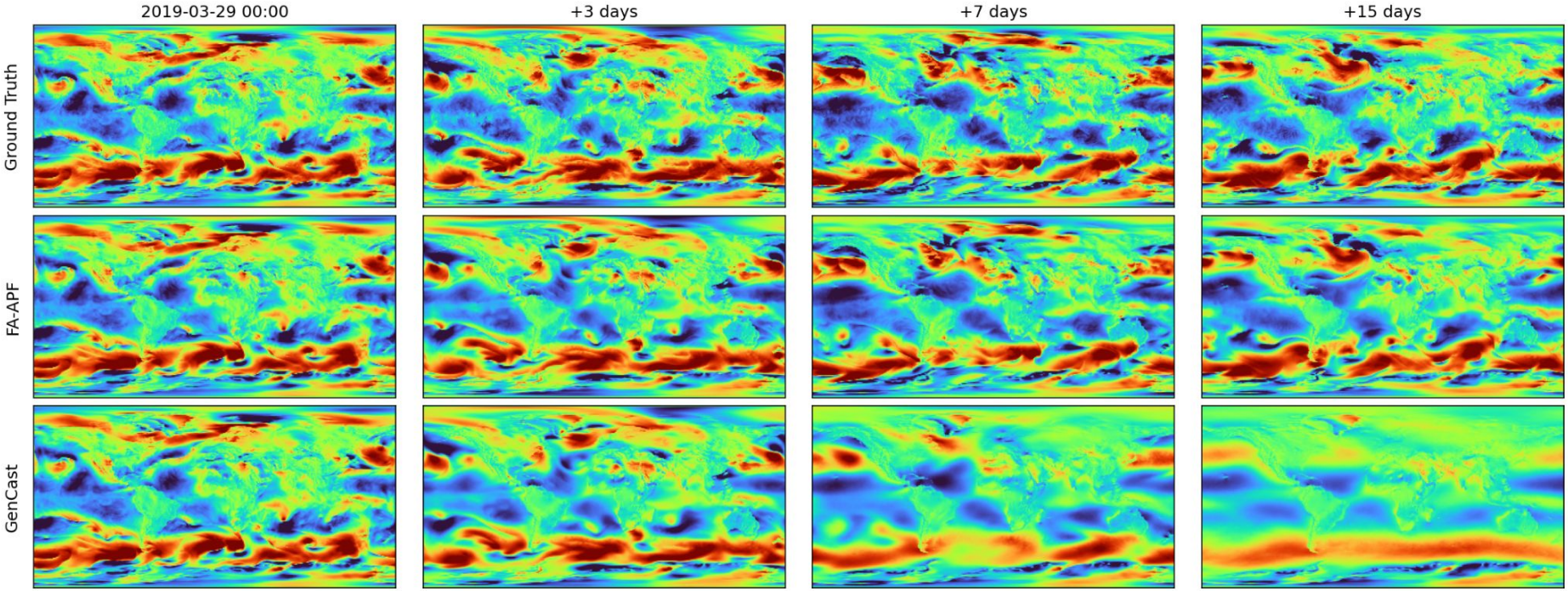


Figure 3. The ensemble mean of the particle filter stays close to the ground truth, even for unobserved variables like the 10m U component of wind.