Training-Free Bayesian Filtering with Generative Emulators

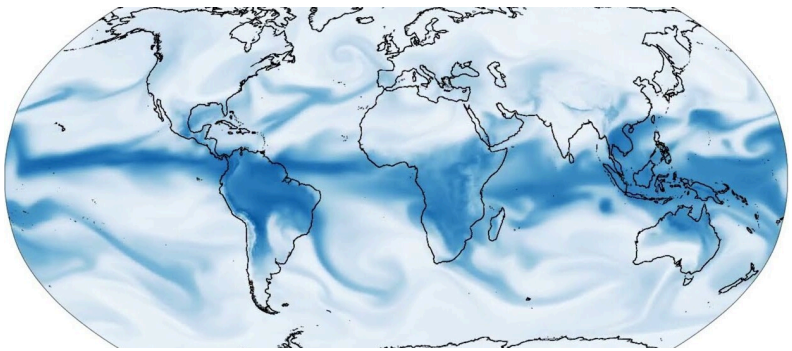
Forty-third International Conference on Machine Learning
Poster Session 5 - Wednesday 8 July 5 p.m. - Poster #1405



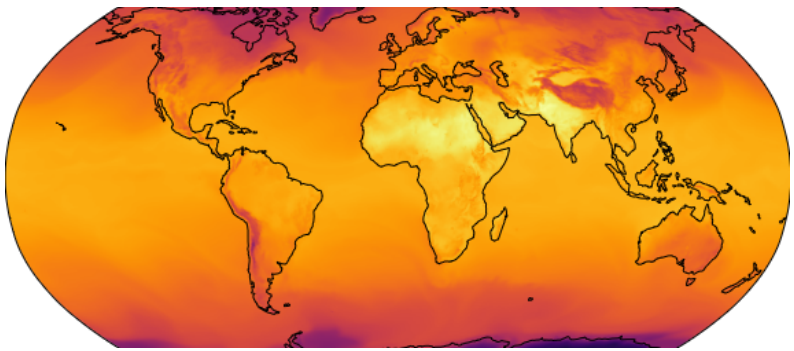
Thomas Savary[1] François Rozet[1] Gilles Louppe[1]

[1]SAIL, Montefiore Institute, University of Liège, Belgium

Over the past decades, weather forecasting has improved dramatically, and forecasts can now be accessed almost instantly from our smartphones.



Specific humidity of the atmosphere at 700 hPa



Surface temperature

These advances have been made possible by continuous **improvements in dynamical models** and **advances in computing capabilities**.

Credits: Google DeepMind

The atmosphere is discretized on a grid, and, **starting from an initial condition**, we solve partial differential equations to simulate its future evolution.

Credits: Meteomatics

The atmosphere is discretized on a grid, and, **starting from an initial condition**, we solve partial differential equations to simulate its future evolution.

⚠️ **Estimating this initial condition is challenging.**

Credits: Meteomatics

The atmosphere is discretized on a grid, and, **starting from an initial condition**, we solve partial differential equations to simulate its future evolution.

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\frac{\nabla p}{\rho} + \nu \nabla^2 u$$

Prior knowledge of the system dynamics

⚠️ **Estimating this initial condition is challenging**. Bayesian filtering addresses this problem by combining **prior knowledge of the dynamics**

Credits: Meteomatics

The atmosphere is discretized on a grid, and, **starting from an initial condition**, we solve partial differential equations to simulate its future evolution.

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\frac{\nabla p}{\rho} + \nu \nabla^2 u$$

Prior knowledge of the system dynamics

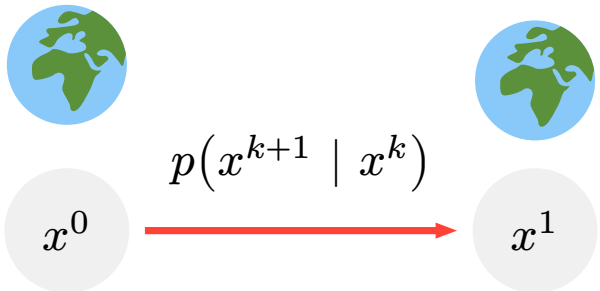
Noisy and sparse observations

⚠️ **Estimating this initial condition is challenging**. Bayesian filtering addresses this problem by combining **prior knowledge of the dynamics** with **observations**.

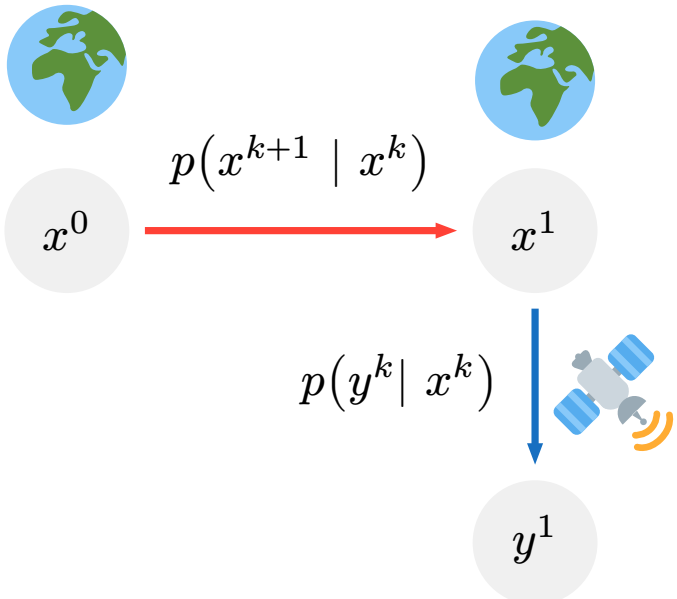
Credits: Meteomatics

x^0

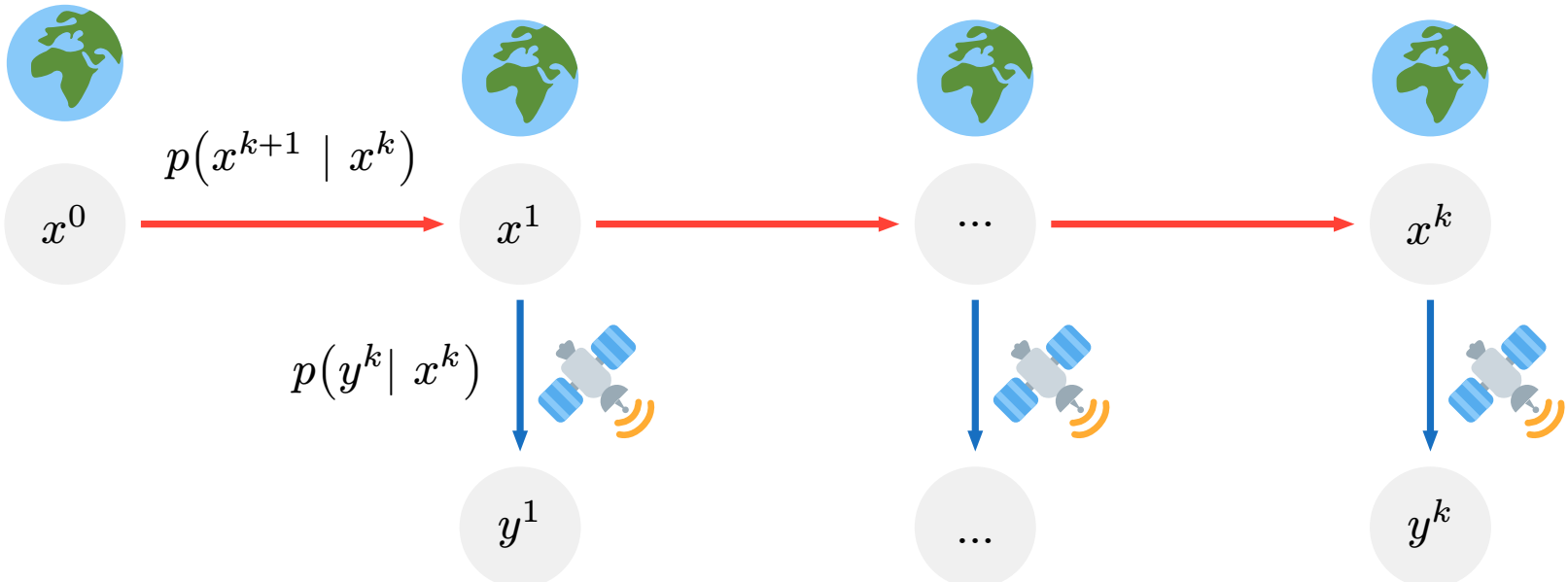
Given a **prior distribution** $p(x_0)$,



Given a **prior distribution** $p(x_0)$, a **Markovian transition model** $p\left(x^{k+1} \mid x^k\right)$

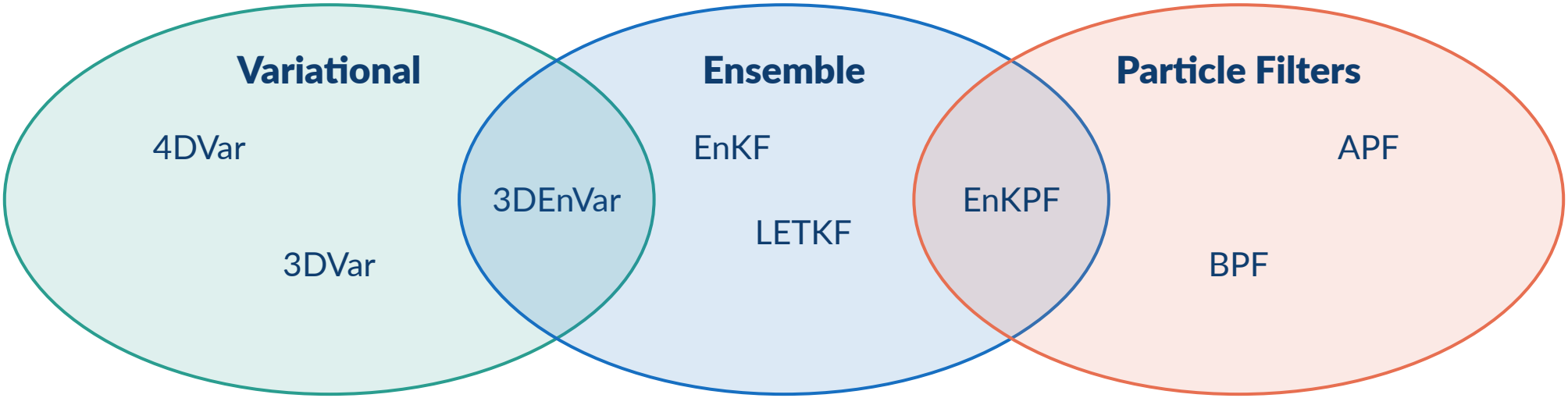


Given a **prior distribution** $p(x_0)$, a **Markovian transition model** $p\left(x^{k+1} \mid x^k\right)$ and an **observation model** $p\left(y^k \mid x^k\right)$,

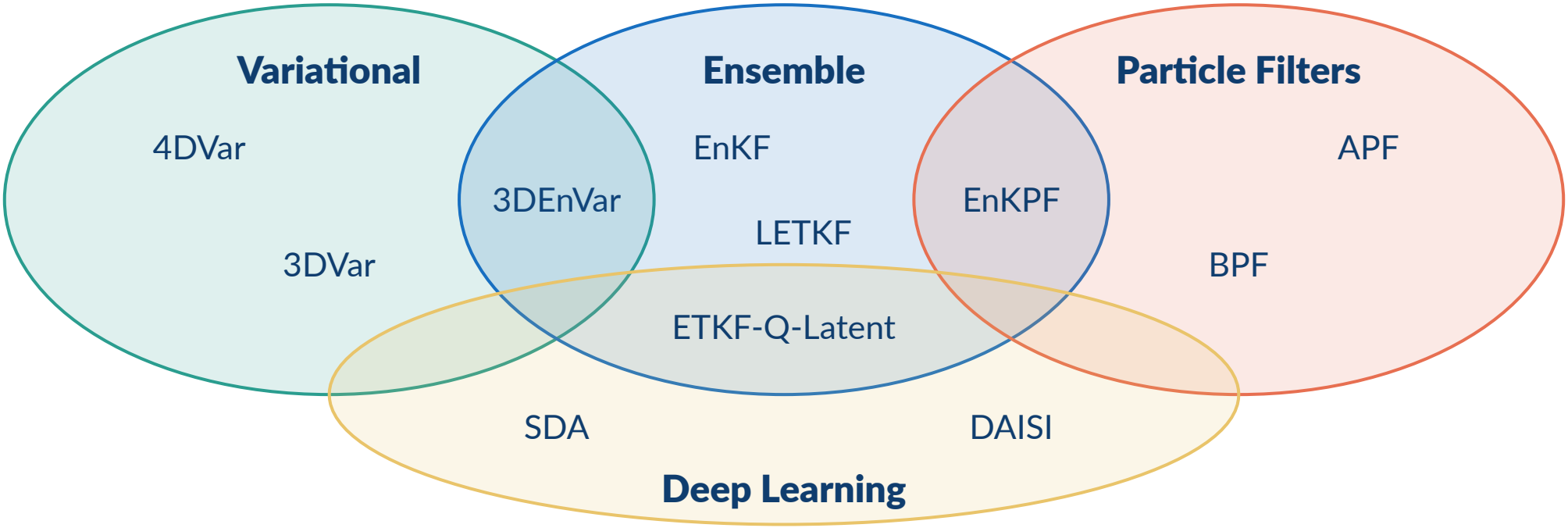


Given a **prior distribution** $p(x_0)$, a **Markovian transition model** $p(x^{k+1} \mid x^k)$ and an **observation model** $p(y^k \mid x^k)$, the goal of **Bayesian filtering** is to estimate plausible states from current and past observations

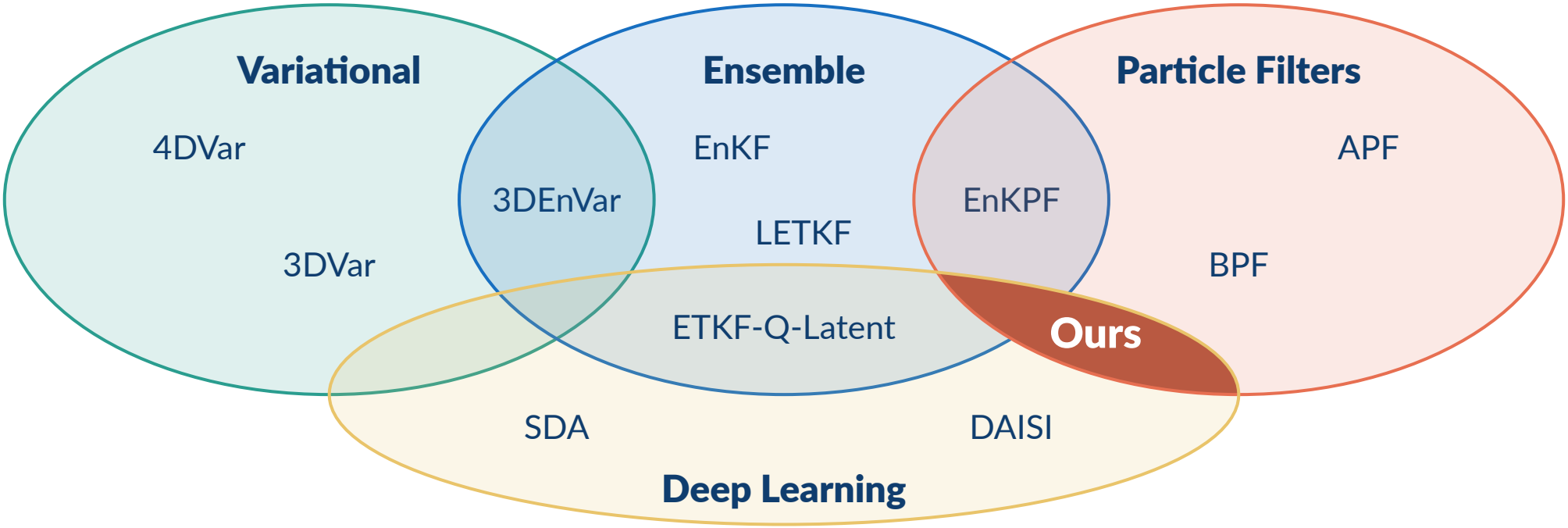
$$p(x^k \mid y^{1:k}) = p(x^k \mid y^1, \dots, y^k) \propto p(y^k \mid x^k) \int p(x^k \mid x^{k-1}) p(x^{k-1} \mid y^{1:k-1}) dx^{k-1}.$$



Credits: Arcucci et al, npj Artificial Intelligence, 2026



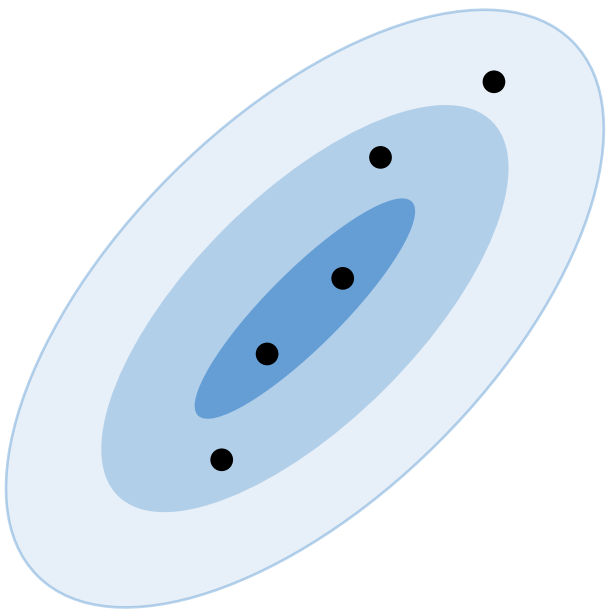
Credits: Arcucci et al, npj Artificial Intelligence, 2026



"Finally, the community anticipates that ML will help address longstanding limitations of traditional DA […]. ***For example, ML may help operationalize fully non-Gaussian DA methods such as particle filters, which have historically been limited by computational cost in high-dimensional systems.***"

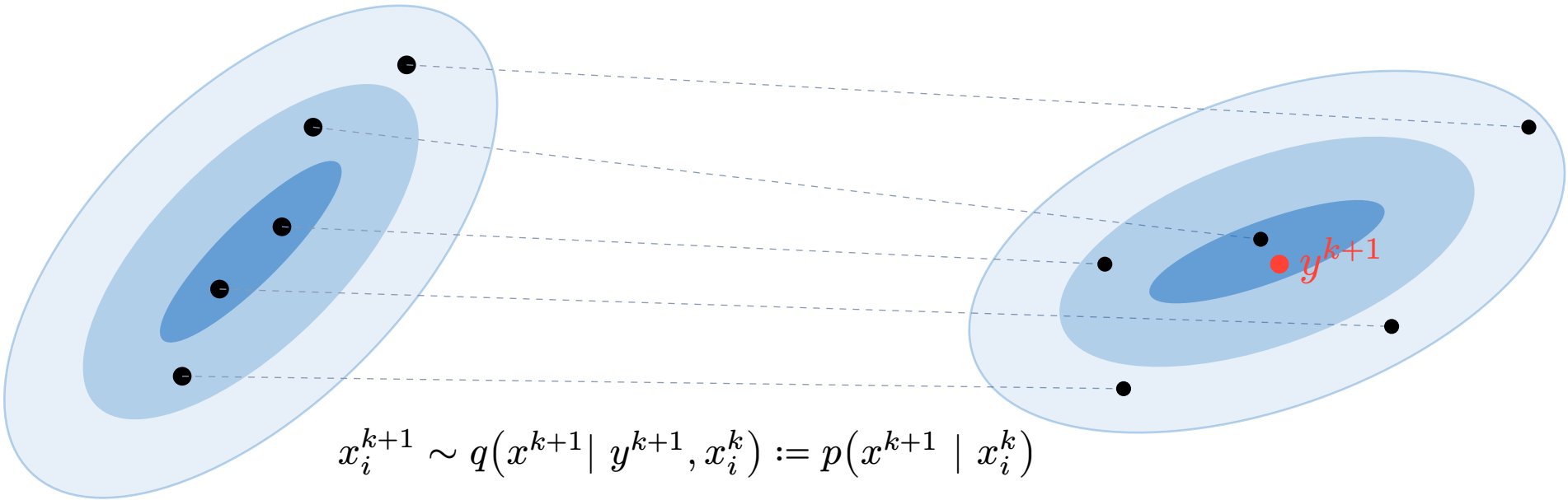
Credits: Arcucci et al, npj Artificial Intelligence, 2026

Particle filters are iterative algorithms that approximate the posterior distribution $p\left(x^k \mid y^{1:k}\right)$ by a discrete probability measure.



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

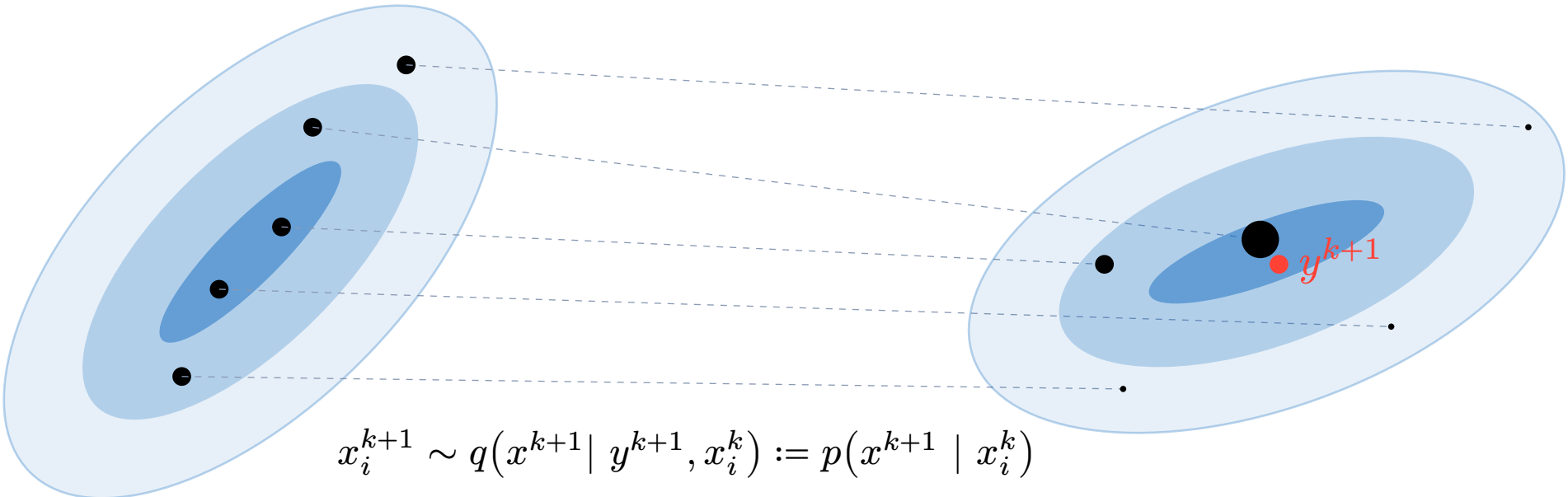
Particle filters are iterative algorithms that approximate the posterior distribution $p\left(x^k \mid y^{1:k}\right)$ by a discrete probability measure. Given a new observation y^{k+1}, particles are propagated using a **proposal distribution** $q\left(x^{k+1} \mid y^{k+1}, x^k\right)$



$$x_i^{k+1} \sim q\left(x^{k+1} \mid y^{k+1}, x_i^k\right) := p\left(x^{k+1} \mid x_i^k\right)$$

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

Particle filters are iterative algorithms that approximate the posterior distribution $p\left(x^k \mid y^{1:k}\right)$ by a discrete probability measure. Given a new observation y^{k+1}, particles are propagated using a **proposal distribution** $q\left(x^{k+1} \mid y^{k+1}, x^k\right)$ and **weighted** to correct the mismatch.

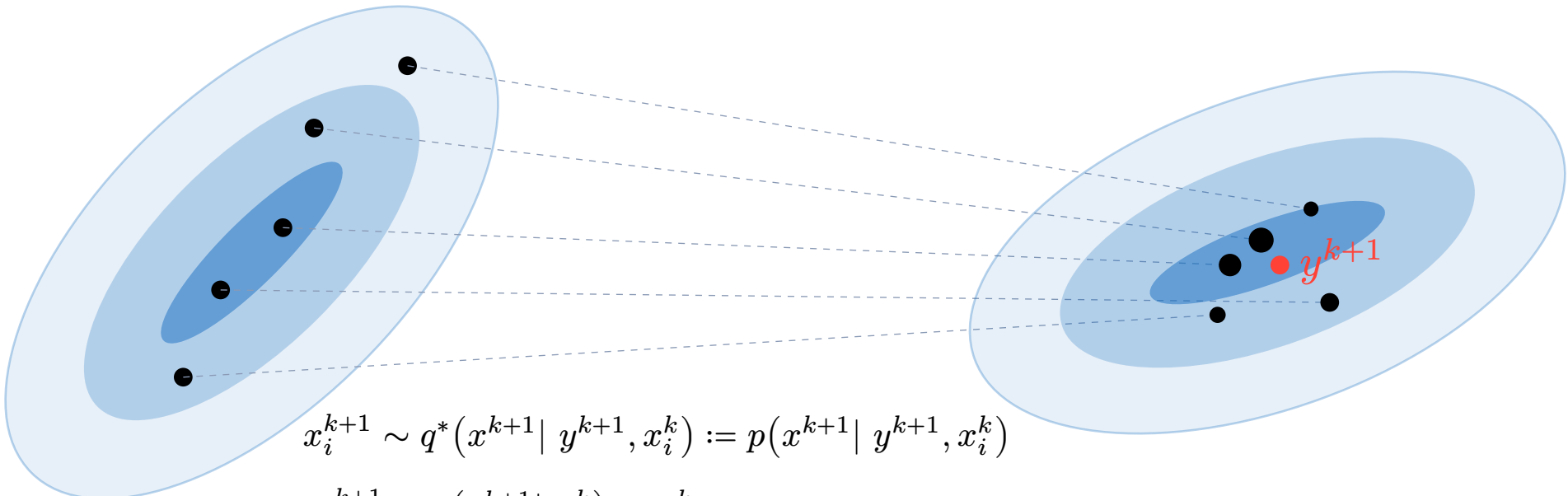


$$x_i^{k+1} \sim q\left(x^{k+1} \mid y^{k+1}, x_i^k\right) := p\left(x^{k+1} \mid x_i^k\right)$$

$$w_i^{k+1} \propto p\left(y^{k+1} \mid x_i^{k+1}\right) \times w_i^k$$

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

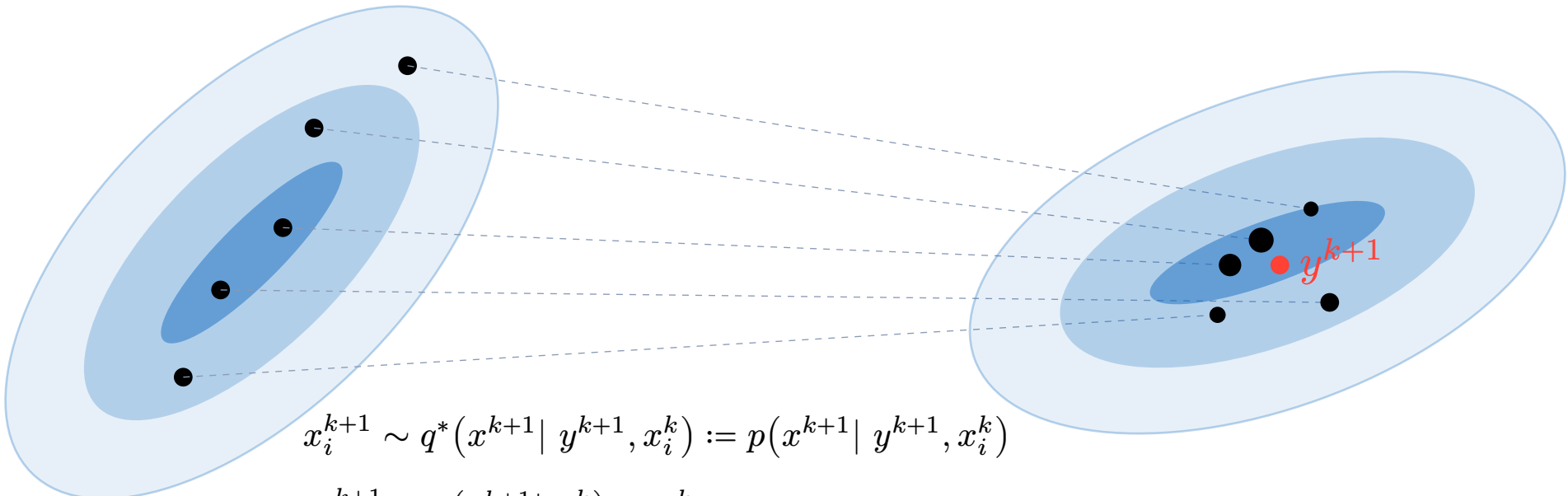
It has been shown in the particle filters literature (Snyder et al, 2015) that the **optimal proposal** $q^*\left(x^{k+1} \middle|\, y^{k+1}, x^k\right) = p\left(x^{k+1} \,\middle|\, y^{k+1}, x^k\right)$ minimizes weights degeneracy.



$$x_i^{k+1} \sim q^*\left(x^{k+1} \middle|\, y^{k+1}, x_i^k\right) := p\left(x^{k+1} \middle|\, y^{k+1}, x_i^k\right)$$

$$w_i^{k+1} \propto p\left(y^{k+1} \middle|\, x_i^k\right) \times w_i^k$$

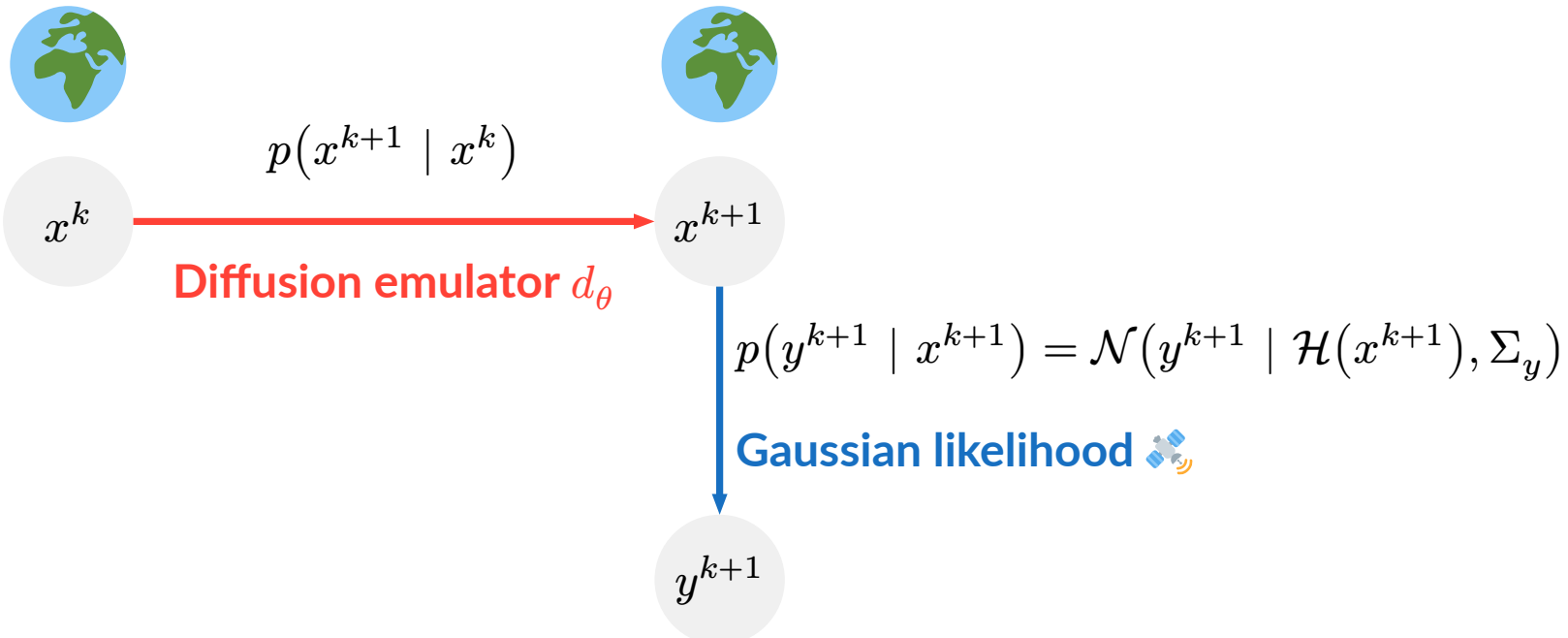
It has been shown in the particle filters literature (Snyder et al, 2015) that the **optimal proposal** $q^*\left(x^{k+1} \middle|\, y^{k+1}, x^k\right) = p\left(x^{k+1} \mid y^{k+1}, x^k\right)$ minimizes weights degeneracy.



$$x_i^{k+1} \sim q^*\left(x^{k+1} \middle|\, y^{k+1}, x_i^k\right) := p\left(x^{k+1} \middle|\, y^{k+1}, x_i^k\right)$$

$$w_i^{k+1} \propto p\left(y^{k+1} \middle|\, x_i^k\right) \times w_i^k$$

⚠️ **It is (very) difficult to sample from the optimal distribution and compute the associated weights for classical numerical solvers**, but (almost) not for diffusion emulators.



💡 Given a **diffusion emulator** for the transition $p(x^{k+1} \mid x^k)$, we can perform approximate **zero-shot sampling from the optimal proposal** $p(x^{k+1} \mid y^{k+1}, x^k)$ (i.e. without further training).

Starting from a noisy sample $x_{t=1}^{k+1}$ and given a particle x_i^k, we can draw sample from the optimal proposal $p\left(x^{k+1} \mid y^{k+1}, x_i^k\right)$ by solving the following SDE from $t=1$ to $t=0$

$$dx_t^{k+1} = \left[f_t x_t^{k+1} - g_t^2 \nabla_{x_t^{k+1}} \log p_t\left(x_t^{k+1} \mid y^{k+1}, x_i^k\right)\right] dt + g_t dw_t.$$

Starting from a noisy sample $x_{t=1}^{k+1}$ and given a particle x_i^k, we can draw sample from the optimal proposal $p\big(x^{k+1} \mid y^{k+1}, x_i^k\big)$ by solving the following SDE from $t=1$ to $t=0$

$$dx_t^{k+1} = \Big[f_t x_t^{k+1} - g_t^2 \nabla_{x_t^{k+1}} \log p_t\big(x_t^{k+1} \mid y^{k+1}, x_i^k\big)\Big] dt + g_t dw_t.$$

The posterior score $\nabla_{x_t^{k+1}} \log p_t\big(x_t^{k+1} \mid y^{k+1}, x_i^k\big)$ is unknown a priori, but can be decomposed via Bayes' rule as

$$\nabla_{x_t^{k+1}} \log p_t\big(x_t^{k+1} \mid x_i^k\big) + \nabla_{x_t^{k+1}} \log p_t\big(y^{k+1} \mid x_t^{k+1}, x_i^k\big).$$

Starting from a noisy sample $x_{t=1}^{k+1}$ and given a particle x_i^k, we can draw sample from the optimal proposal $p\big(x^{k+1} \mid y^{k+1}, x_i^k\big)$ by solving the following SDE from $t=1$ to $t=0$

$$dx_t^{k+1} = \Big[f_t x_t^{k+1} - g_t^2 \nabla_{x_t^{k+1}} \log p_t\big(x_t^{k+1} \mid y^{k+1}, x_i^k\big)\Big] dt + g_t dw_t.$$

The posterior score $\nabla_{x_t^{k+1}} \log p_t\big(x_t^{k+1} \mid y^{k+1}, x_i^k\big)$ is unknown a priori, but can be decomposed via Bayes' rule as

$$\nabla_{x_t^{k+1}} \log p_t\big(x_t^{k+1} \mid x_i^k\big) + \nabla_{x_t^{k+1}} \log p_t\big(y^{k+1} \mid x_t^{k+1}, x_i^k\big).$$

The first term is already known using the trained denoiser, and following Rozet et al. (NeurIPS 2024), we approximate the second term by

$$\nabla^T_{x_t^{k+1}} \mathbb{E}\big[x^{k+1} \mid x_t^{k+1}, x_i^k\big] H^T \big(\Sigma_y + H \mathbb{V}\big[x^{k+1} \mid x_t^{k+1}, x_i^k\big] H^T\big)^{-1} \big(y - H \mathbb{E}\big[x^{k+1} \mid x_t^{k+1}, x_i^k\big]\big).$$

⚠️ Computing particle weights for the optimal proposal is not trivial since we need to evaluate

$$w_i^{k+1} \propto p\left(y^{k+1} \mid x_i^k\right) = \int p\left(y^{k+1} \mid x^{k+1}\right) p\left(x^{k+1} \mid x_i^k\right) dx^{k+1}.$$

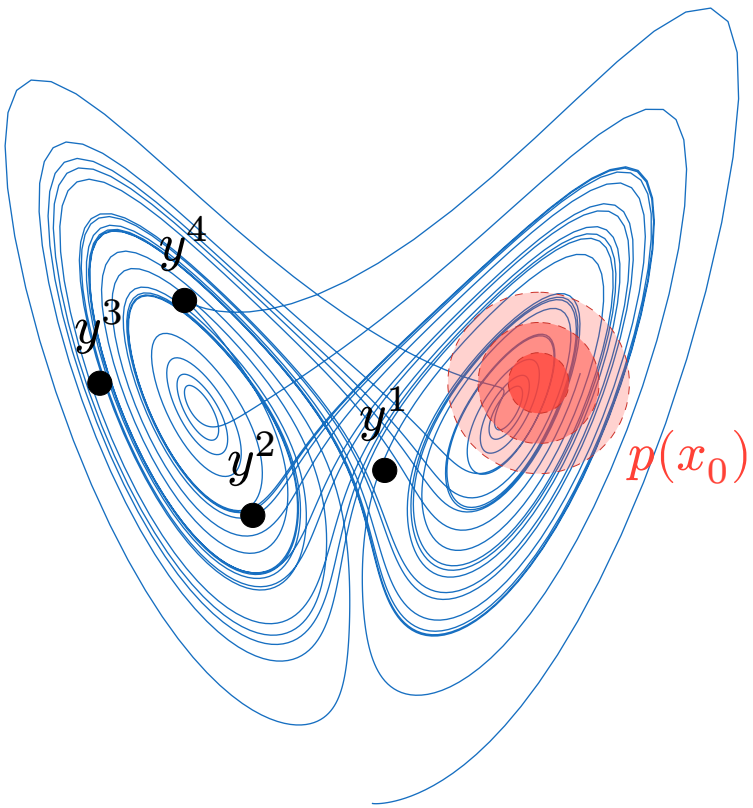
Assuming that the transition distribution $p\left(x^{k+1} \mid x^k\right)$ is a Dirac at $\mathbb{E}\left[x^{k+1} \mid x_i^k\right]$, we obtain

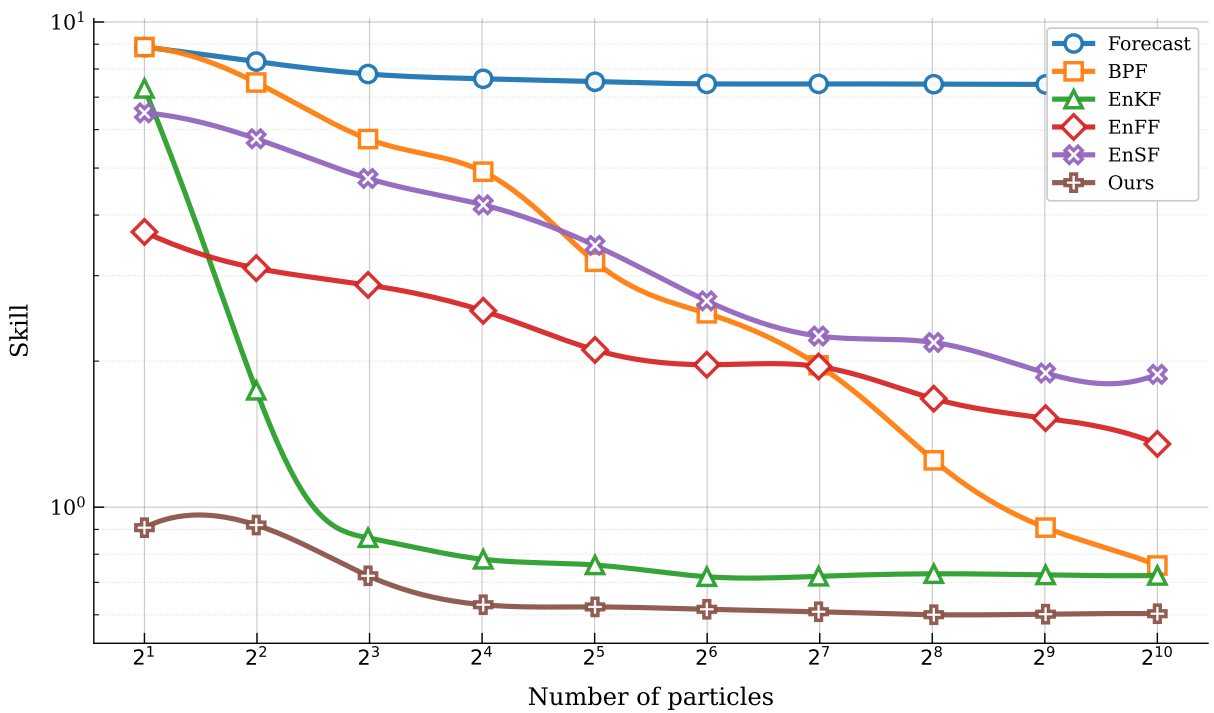
$$w_i^{k+1} \propto \| y^{k+1} - \mathcal{H}\left(\mathbb{E}\left[x^{k+1} \mid x_i^k\right]\right) \|_{\alpha^{-1}\Sigma_y},$$

and we can compute the expectation of the next state with only one call to the denoiser

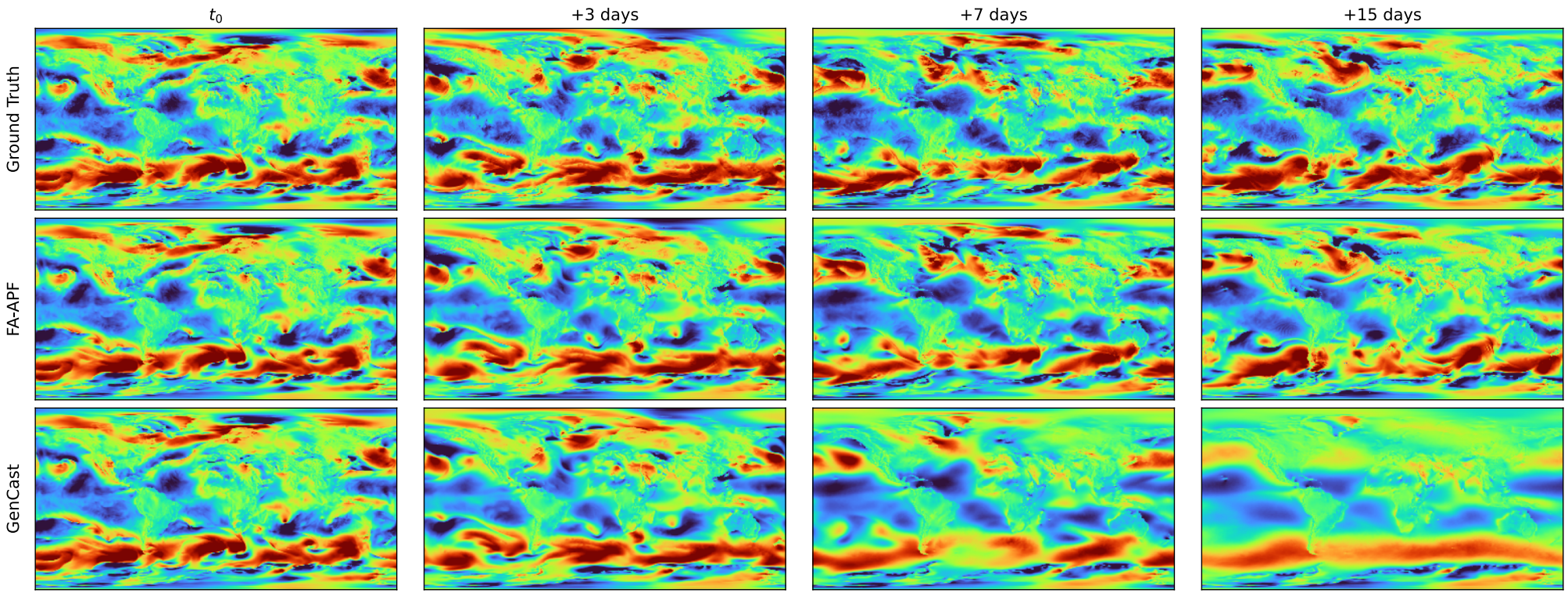
$$\mathbb{E}\left[x^{k+1} \mid x_i^k\right] \underset{\varepsilon \sim \mathcal{N}(0,I)}{=} \mathbb{E}\left[x^{k+1} \mid x_i^k, \sigma_1 \varepsilon\right] \approx d_\theta\left(t = 1, x^k = x_i^k, x_t^{k+1} = \sigma_1 \varepsilon\right).$$

We first evaluate our approach on a stochastic version of the **Lorenz-63** system (Chekroun et al, 2011). **Our method is more effective than the BPF**, especially with a small number of particles.





We also applied our method to the atmosphere using the pre-trained GenCast (Price et al, 2023) denoiser, enabling the assimilation of sparse observations inspired by real stations.



Key Takeaways

- Particle filters are **exact in nonlinear regimes**, but suffer from **degeneracy**.
- **Diffusion emulators** enable sampling from the **optimal proposal** without further training.
- Experiments on large-scale dynamical systems demonstrate the potential of this method.

Limitations

- In theory, **particle filters still require a large number of particles** in large-scale systems.
- **Weights are approximated**, and the observation covariance Σ_y is **artificially inflated**.
- Zero-shot posterior sampling with diffusion emulators **is not exact**.

Thank you for your attention!

Paper

Code

LinkedIn

See you at the poster session 🙂
Poster #1405