Deep Bayesian Filter for Bayes-Faithful data assimilation

Yuta Tarumi (RIKEN, yuta.tarumi@riken.jp)

Motivation: non-Gaussian data assimilation

Research problem: Estimate the hidden physical state X_t by fusing noisy, incomplete observations with a dynamical-model forecast inside a Bayesian update.

Why it matters: Estimating the true system state from observations is a common, fundamental task across the natural sciences: it powers daily weather prediction and are becoming equally vital in oceanography, hydrology, and other sciences.

Key challenges:
- Observations are noisy and incomplete.
- Non-linear dynamics and observation maps create non-Gaussian posteriors.
- Million-dimensional states make particle-filter weights collapse.

Our strategy: deploy a Variational Autoencoder (VAE) with a **linear latent transition:** the linear step allows closed-form Bayesian updates, while neural encoders/decoders provide the needed non-Gaussianity for the posterior.

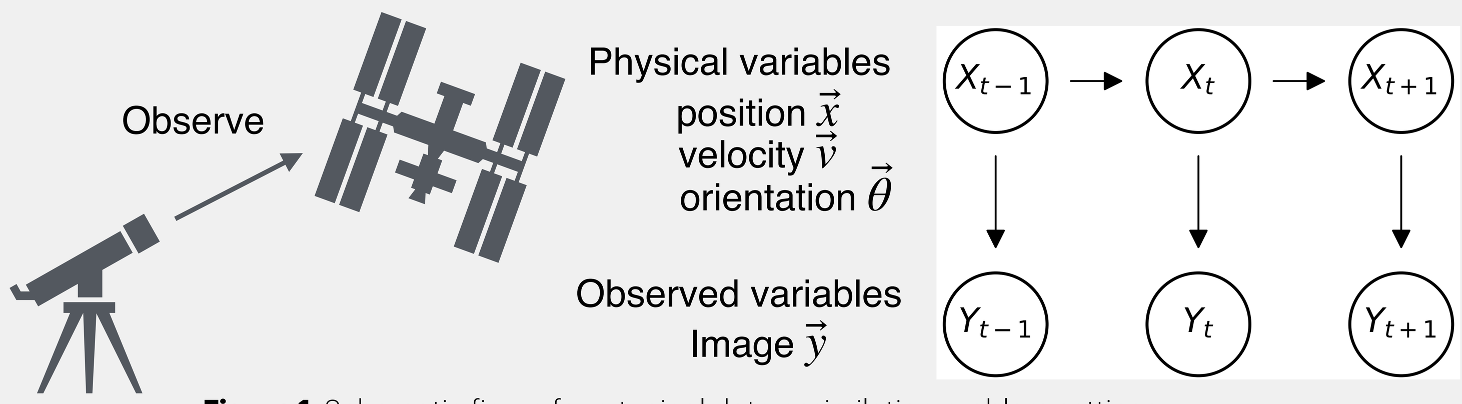

Figure 1: Schematic figure for a typical data assimilation problem settings

Background: non-linear data assimilation

DA in a SSM

A state space model has two models:

Dynamics model: $p(X_t \mid X_{t-1}) = \mathcal{N}[X_t; f(X_t), Q]$

Observation model: $p(Y_t \mid X_t) = \mathcal{N}[Y_t; h(X_t), R]$

In sequential filtering,

Predict step: $p(X_{t+1} \mid Y_{1:t}) = \int p(X_t \mid Y_{1:t}) p(X_{t+1} \mid X_t) dX_t$

Update step: $p(X_{t+1} \mid Y_{1:t+1}) = \dfrac{p(X_{t+1} \mid Y_{1:t}) p(Y_{t+1} \mid X_{t+1})}{p(Y_{t+1} \mid Y_{1:t})}$

Assuming that $p(X_t \mid Y_{1:t})$ is Gaussian,
$p(X_{t+1} \mid Y_{1:t})$ is Gaussian if f is linear.
Assuming that $p(X_{t+1} \mid Y_{1:t})$ is Gaussian,
$p(X_{t+1} \mid Y_{1:t+1})$ is Gaussian if h is linear.

→ **Nonlinearity in f or h results in non-Gaussian posterior distribution**

Method

Our strategy: "Expressive but computationally tractable posterior class"

- **Gaussian posterior over the new latent variables h_t**
  - **Linear latent dynamics** so that the Gaussianity is conserved in the predict step
  - **Inverse observation operator** $r(h_t \mid Y_t)$ for the update step to keep Gaussianity

  with these, posterior remains always Gaussian: **computationally tractable**
- **Nonlinear emission model** to ensure representation power over X_t

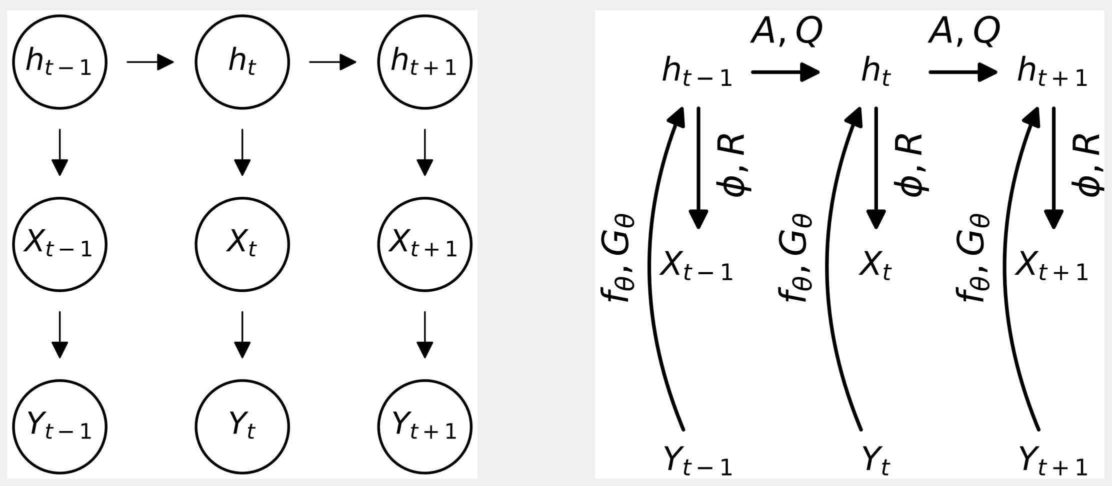

Figure 2: left panel: graphical model of our method. Right panel: inference structure.

New SSM

Dynamics model: $p(h_t \mid h_{t-1}) = \mathcal{N}[h_t; Ah_t, Q]$: **Linear dynamics**

Emission model: $p(X_t \mid h_t) = \mathcal{N}[X_t; \phi(h_t), R]$: **nonlinear emission**

With these new models,

Predict step: $p(h_{t+1} \mid Y_{1:t}) = \int p(h_t \mid Y_{1:t}) p(h_{t+1} \mid h_t) dh_t$

Update step: $p(h_{t+1} \mid Y_{1:t+1}) = \dfrac{r(h_{t+1} \mid Y_{t+1})}{\rho(h_t)} p(Y_{t+1} \mid X_{t+1})$

$$\rho(h_t) = \mathcal{N}(h_t; m, V)$$

$$r(h_t \mid Y_t) = \frac{\rho(h_t) p(Y_t \mid h_t)}{\int \rho(h_t) p(Y_t \mid h_t) dh_t} = \mathcal{N}[h_t; f_\theta(Y_t), G_\theta(Y_t)]$$

Filtered distribution over X_t: $p(X_t \mid Y_{1:t}) = \int p(h_t \mid Y_{1:t}) p(X_t \mid h_t) dh_t$

Training objective

Joint ELBO: $\int q(h_t \mid Y_{1:t}) \dfrac{p(h_t, X_t, Y_t \mid X_{1:t-1}, Y_{1:t-1})}{q(h_t \mid Y_{1:t})} \leq \log p(X_t, Y_t \mid X_{1:t-1}, Y_{1:t-1})$

Trainable components	Features
IOO(encoder) $r(h_t \mid Y_t)$	Encoder in VAE, extracts information from observations
decoder $\phi(h_t)$	Decoder in VAE, maps latent h_t to the physical variables X_t
Transition matrix A	Parametrized as block diagonal, see below
Emission noise R	Trained to represent the model confidence on X_t prediction

Compute-efficient parametrization

We assume block-diagonal form for A

Even with a large number of (λ_i, θ_i), high-dimensional variables, matrix multiplication & inversions are quick

$$A = \begin{pmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & A_n \end{pmatrix}$$

$$A_i = e^{\lambda_i} \begin{pmatrix} \cos\theta_i, & -\sin\theta_i \\ \sin\theta_i, & \cos\theta_i \end{pmatrix}$$

Our Deep Bayesian Filter builds on the dynamical VAE (Girin et al., 2021): **linear-Gaussian latent dynamics yield closed-form Bayesian updates, while the nonlinear decoder keeps the posterior over physical variables expressive**

Training stability

Training RNN-based DVAEs is often unstable because their gradients can explode or vanish.

Our DBF instead advances the state with a fixed matrix; as long as its eigenvalues stay near or below one, predictions and losses remain bounded.

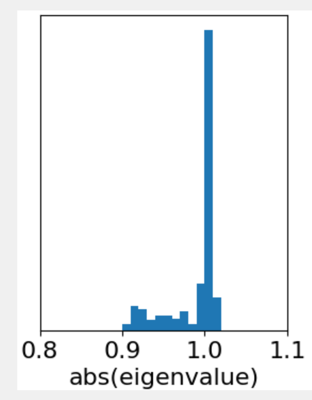

Figure 3: an example histogram for abs(eigenvalue) of the dynamics matrix A trained for a Lorenz96 experiment

Experimental results

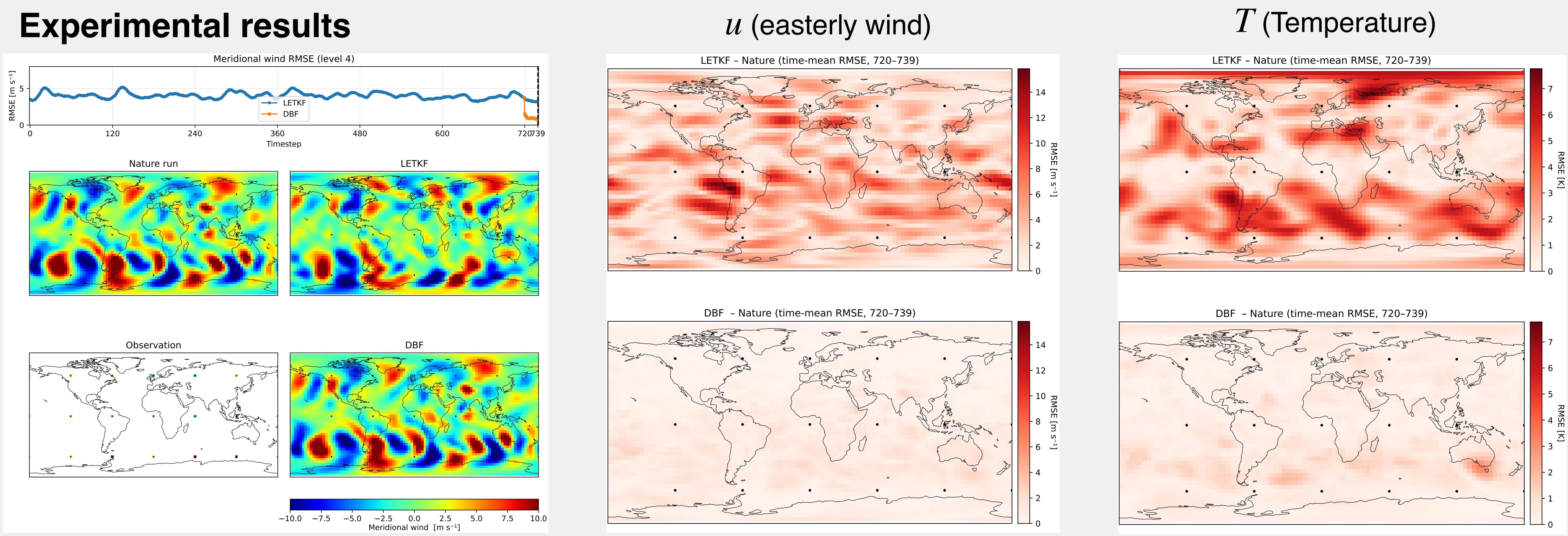

Figure 4: left panel compares the nature run (top left), observation stations (bottom left), LETKF inference result (top right), and DBF inference result (bottom right). Center and right panels show the inference RMSE against the nature run, averaged over the assimilation window (5 days). Center panel shows u and the right panel shows T.

Model: SPEEDY atmospheric model. $8 \times 48 \times 96$ grid, u (east-west wind), v (north-south wind), T (temperature), q (water content), p_s (surface pressure)

Architecture: 2048 latent variable dimensions, transformer for encoder ($Y \to h$) and decoder ($h \to X$)

Results: Even with $8 \times 3 \times 6$ observation grid (0.4 % of all grids), our method successfully recover the atmospheric state

Future work: Higher ($\sim 10^9$) dimension problems, or an obs-to-obs application like Aardvark/AI-DOP?

Scan me for our code!

Scan me for our paper!

Reference
Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber and Xavier Alameda-Pineda (2021), "Dynamical Variational Autoencoders: A Comprehensive Review", Foundations and Trends® in Machine Learning: Vol. 15: No. 1-2, pp 1-175. http://dx.doi.org/10.1561/2200000089