Ensemble DA emulation with Generative AI

Training Course: DA & ML

Wei Pan

wei.pan@ecmwf.int

ECMWF

Lecture roadmap

- Data assimilation as probabilistic inference
- Generative AI
  - Evolution of GenAI
  - Mathematical formulation: learning probability distributions
  - Parametric density estimation
  - Examples of modern techniques
- EDA and generative methods
  - EDA statistics emulator (experimental)
  - EDA ensemble emulator (in development)
- Additional mathematical details in the appendix

Bayesian data assimilation

Bayes' theorem

$$\underbrace{p(\mathbf{x} \mid \mathbf{y})}_{\text{posterior}} \propto \underbrace{p(\mathbf{y} \mid \mathbf{x})}_{\text{likelihood}} \underbrace{p(\mathbf{x})}_{\text{prior}}.$$

In our DA context, under Gaussian background, model error and observation error assumptions, the *negative log posterior* (weak constraint 4D-Var cost function) becomes

$$\begin{aligned} \underbrace{-\log p(\mathbf{x}_{0:K} \mid \mathbf{y}_{1:K})}_{\text{log posterior}} = & \underbrace{\tfrac{1}{2}(\mathbf{x}_0 - \mathbf{x}_b)^\top \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_b)}_{\text{background}} \\ & + \underbrace{\tfrac{1}{2}\sum_{k=1}^{K}(\mathbf{x}_k - \mathcal{M}_{k-1\to k}(\mathbf{x}_{k-1}))^\top \mathbf{Q}_k^{-1}(\mathbf{x}_k - \mathcal{M}_{k-1\to k}(\mathbf{x}_{k-1}))}_{\text{model error likelihood}} \\ & + \underbrace{\tfrac{1}{2}\sum_{k=1}^{K}(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))^\top \mathbf{R}_k^{-1}(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))}_{\text{observation likelihood}} + \text{const.} \end{aligned}$$

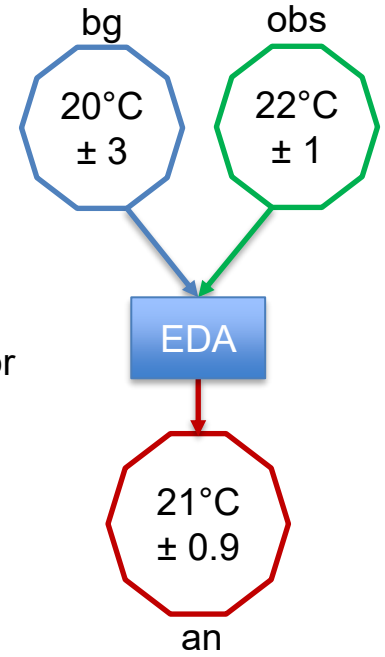
Ensemble data assimilation (EDA)

Variational DA produces the *maximum a posterior* (MAP) estimate

$$\mathbf{x}^a = \arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{y}).$$

EDA implements an *approximate sampling mechanism* by running many perturbed MAP optimisations,

$$\mathbf{x}^{a,(i)} = \arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{y}^{(i)}, \mathbf{x}^{b,(i)}, \boldsymbol{\eta}^{(i)}), \quad i = 1, \ldots, N.$$



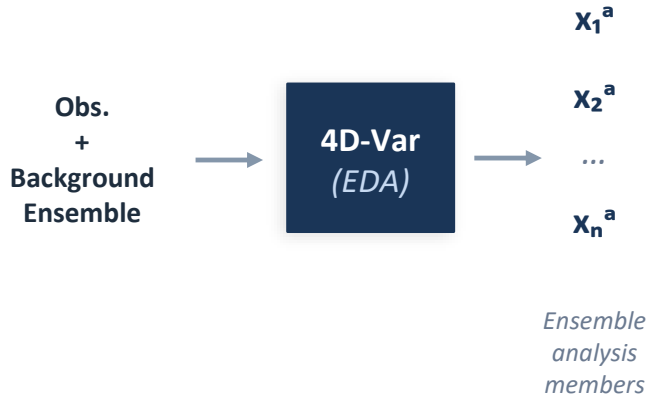
The ensemble covariance provides a flow-dependent estimate of the analysis covariance (or the background error covariance $\boldsymbol{B}$ when in forecast mode),

$$\mathbf{P}^a = \frac{1}{N-1} \sum_{i=1}^{N} \left(\mathbf{x}^{a,(i)} - \bar{\mathbf{x}}^a\right)\left(\mathbf{x}^{a,(i)} - \bar{\mathbf{x}}^a\right)^T \approx \mathrm{Cov}(\mathbf{x} \mid \mathbf{y}).$$

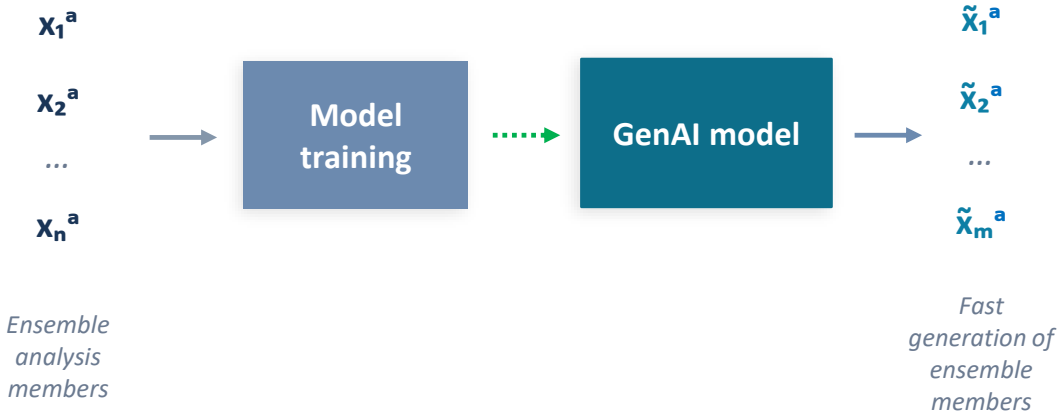
The EDA ensemble distribution can exhibit non-Gaussian structure, arising from nonlinear model dynamics and nonlinear observation operators.

Motivation for generative models

- Each MAP optimisation requires expensive model linearisation and adjoint integrations.



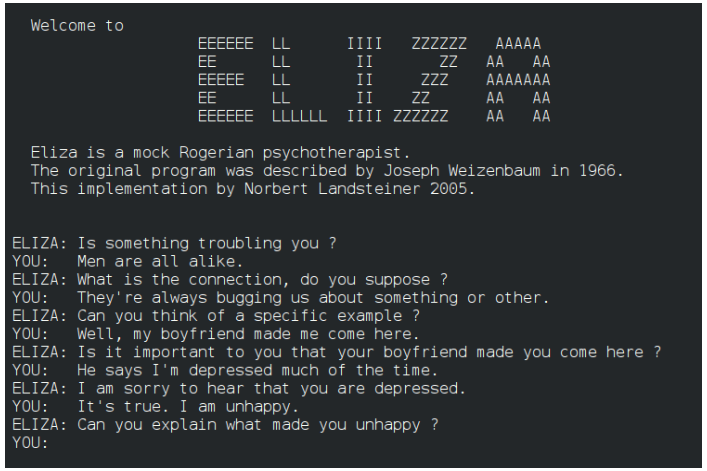
- Generative models offer a potential way to learn the distribution of the ensemble states and generate ensemble members much more efficiently.



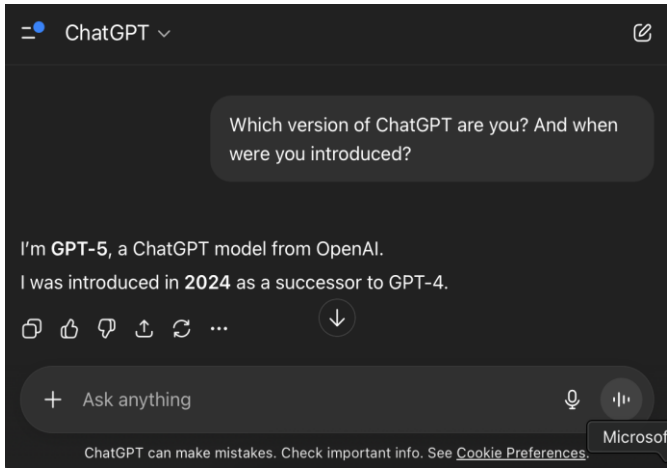
Generative AI

Generative AI refers to AI systems that can produce new content – such as text, images, music, code, or video – based on the data they were trained on.

For example, Large Language Models (LLMs) generate text by repeatedly predicting the next most likely token (such as a word or sub-word).



ELIZA (1966) – first chatbot



GPT-5 (2024)

ChatGPT Gemini LLAMA 4

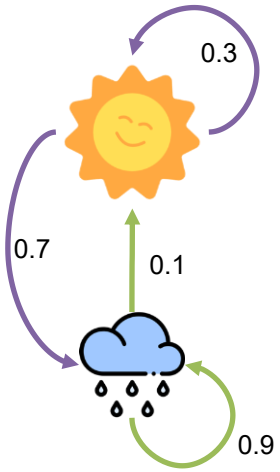
Claude deepseek

However, it's not really GenAI.

Evolution of Generative AI

1913 — Markov Chain

Andrey Markov introduced and applied the idea of a "chain of linked events" to Pushkin's novel "Eugene Onegin" – obtained transition probabilities between vowels/consonants.



1970s — Hidden Markov Model

In particular, the Baum-Welch Expectation Maximisation algorithm for learning probabilistic generative models for speech processing and bioinformatics.

1983 — Boltzmann machine

(Hinton - Nobel prize 2024) Energy-based probabilistic neural network. Basis for later deep generative models.

1994 — Mixture Density Networks

Introduced by C. Bishop in the paper "Mixture Density Networks". A class of neural networks that learns complex multi-modal conditional density from data.

1997 — (LSTM) Long Short Term Memory

Played a foundational role in the development of modern generative AI for sequence data.

2013 — (VAE) Variational Autoencoder

Introduced by D.P. Kingma and M. Welling. First scalable deep latent-variable generative model with probabilistic inference.

2014 — (GAN) Generative Adversarial Networks

Goodfellow et al made a major breakthrough in using adversarial training to produce realistic images.

2015 — Diffusion

The paper "Deep Unsupervised Learning using Nonequilibrium Thermodynamics" introduced the core concept of adding noise to data and then learning to reverse the process to generate new data.

2017 — Transformers

The Transformer model is introduced in the paper "Attention Is All You Need". Foundation for today's LLMs.

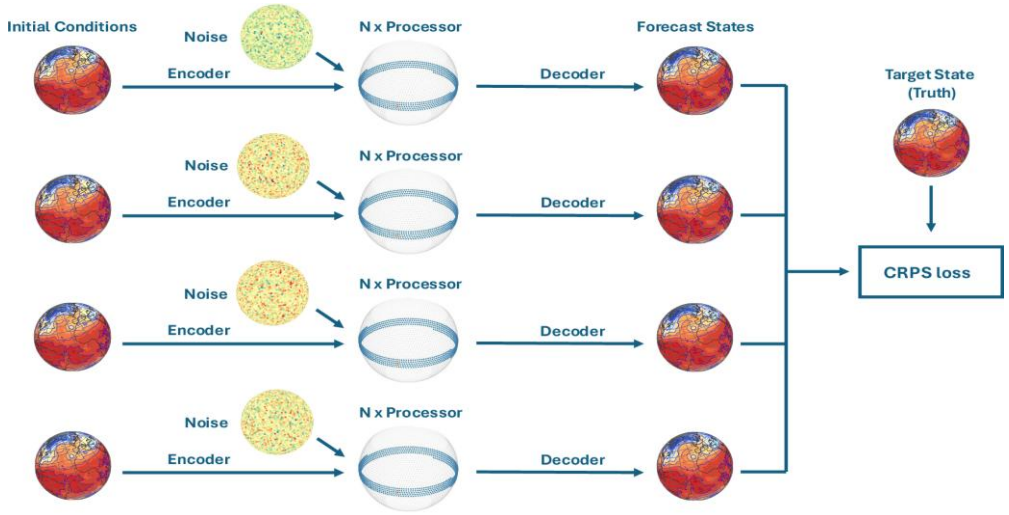
2018 — (GPT) Generative Pretrained Transformer and beyond

LLMs, DDPM, flows etc, etc, etc, etc.…

ECMWF

Example GenAI applications in weather

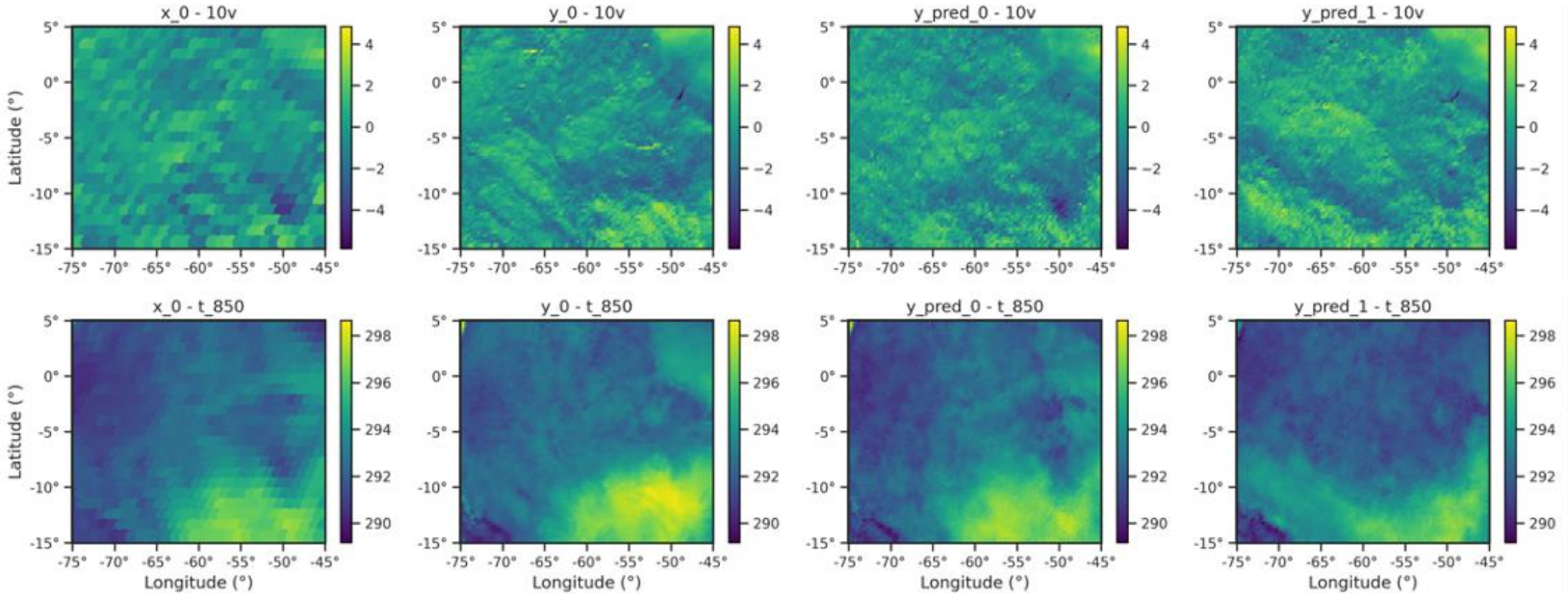
- Probabilistic forecasting



"... a distinct set of random numbers is drawn for each model instance and at each forecast step throughout the forecast... "

AIFS-CRPS, Lang et al (2026), npj

- Downscaling



Amazonian rainforest – 2023.08.01

o96 --> o320

from *Joffrey Dumont Le Brazidec* (ECMWF)

What is Generative AI?

- Given data $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N} \sim p_{\text{data}}(\mathbf{x})$, learn parameters θ of a neural network such that

$$p_\theta(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x}).$$

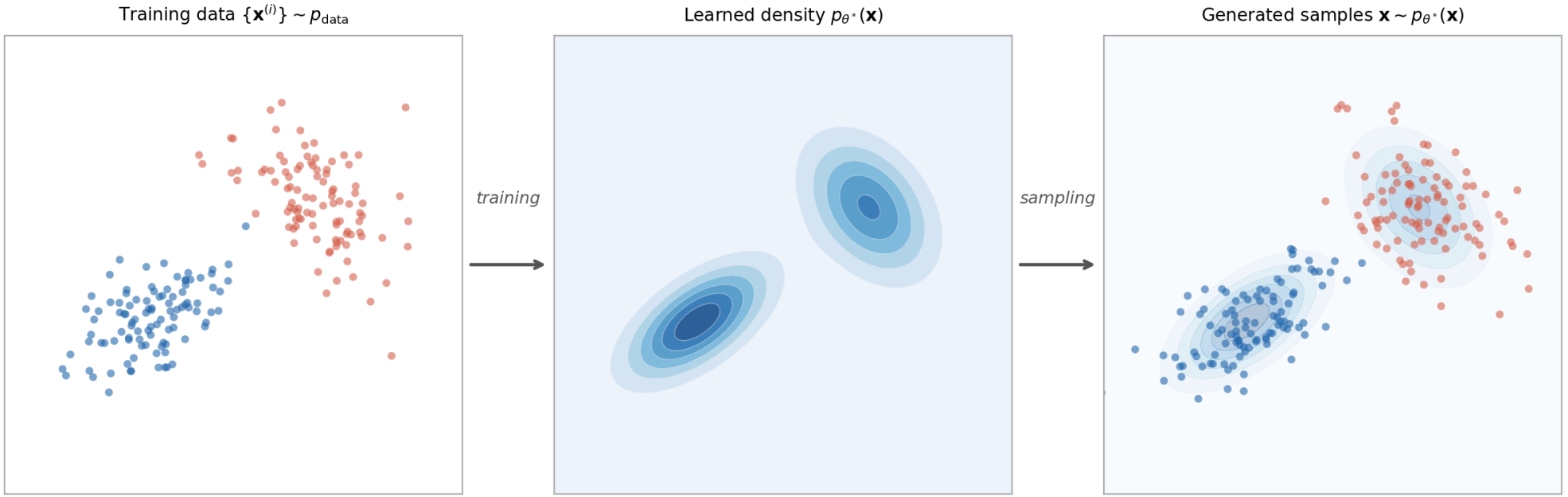
- Learning meaning finding θ that minimises some divergence/discrepancy measure D

$$\theta^* = \arg\min_\theta D(p_{\text{data}} \,\|\, p_\theta).$$

- Draw new samples efficiently at inference time

$$\mathbf{x} \sim p_{\theta^*}(\mathbf{x}).$$

Intuition: *find parameters so that the generative model matches the data distribution as closely as possible w.r.t. D.*



Measuring discrepancy: Kullback-Leibler (KL) divergence

KL divergence, D_{KL}, (a.k.a. *relative entropy* from Shannon's information theory, see Appendix A.1)

$$
\begin{aligned}
D_{\mathrm{KL}}(p_{\mathrm{data}} \,\|\, p_\theta) &:= \mathbb{E}_{p_{\mathrm{data}}}\left[\log \frac{p_{\mathrm{data}}(\mathbf{x})}{p_\theta(\mathbf{x})}\right] \\
&= \mathbb{E}_{p_{\mathrm{data}}}\left[\log p_{\mathrm{data}}(\mathbf{x}) - \log p_\theta(\mathbf{x})\right].
\end{aligned}
$$

Intuition: How much better the true model explains the data compared to p_θ.

D_{KL} is *always non-negative* (see Appendix A.2), and *equals 0* if and only if $p_{\mathrm{data}} = p_\theta$ *almost surely*. It is thus effective as a loss function for optimisation,

$$
\arg\min_\theta D_{\mathrm{KL}}(p_{\mathrm{data}} \,\|\, p_\theta) = \arg\max_\theta \mathbb{E}_{p_{\mathrm{data}}}[\log p_\theta(\mathbf{x})]
$$

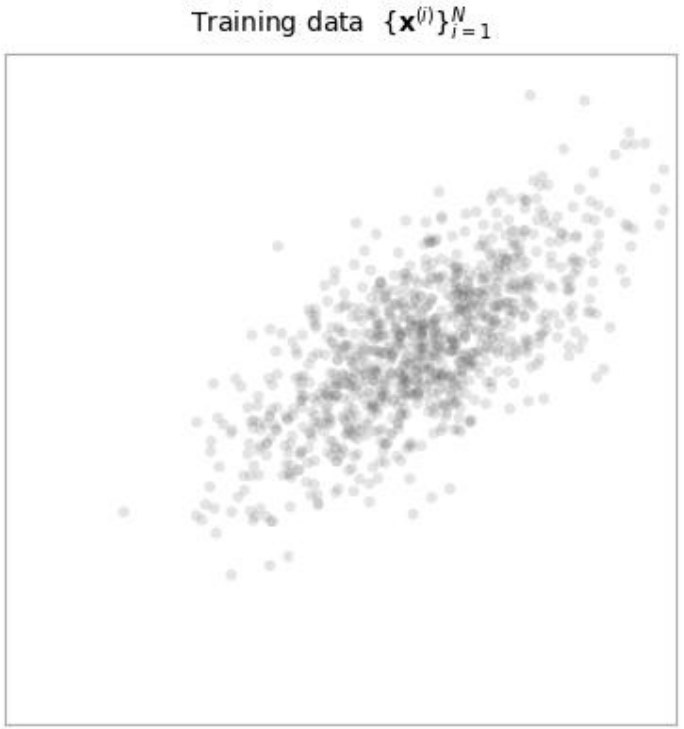
**right hand side is exactly Maximum Likelihood Estimation (MLE)!* 😲 🤯

Given only data samples $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N} \sim p_{\mathrm{data}}(\mathbf{x})$

$$
\arg\max_\theta \mathbb{E}_{p_{\mathrm{data}}}[\log p_\theta(\mathbf{x})] \approx \arg\max_\theta \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)}) \qquad \text{(by the law of large numbers as } N \to \infty)
$$

Example: parametric density model

Choose Gaussian p_θ with $\theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$



1000 data samples

Example: parametric density model

Choose Gaussian p_θ with $\theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

Closed-form MLE for Gaussian (see Appendix B)

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)}, \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} \left(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}\right)\left(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}\right)^\top$$

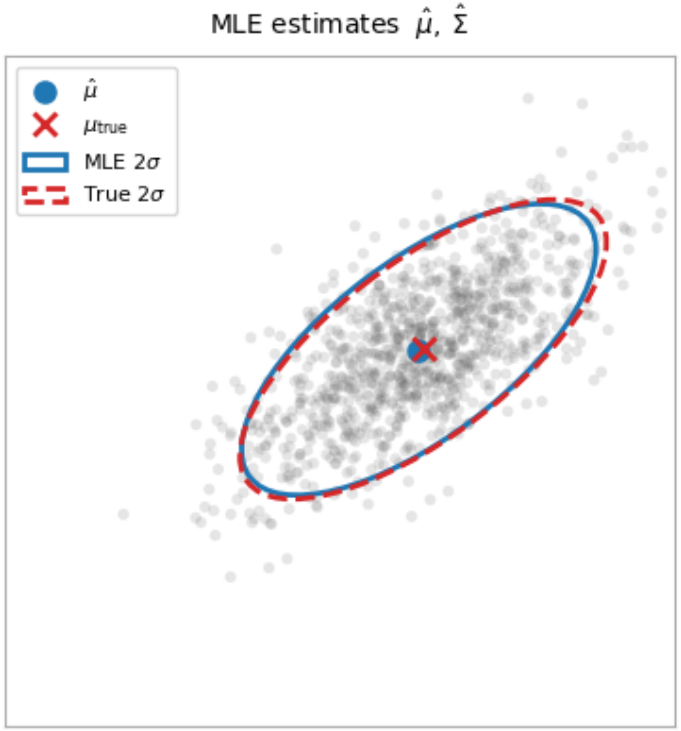
MLE result (1000 data points):

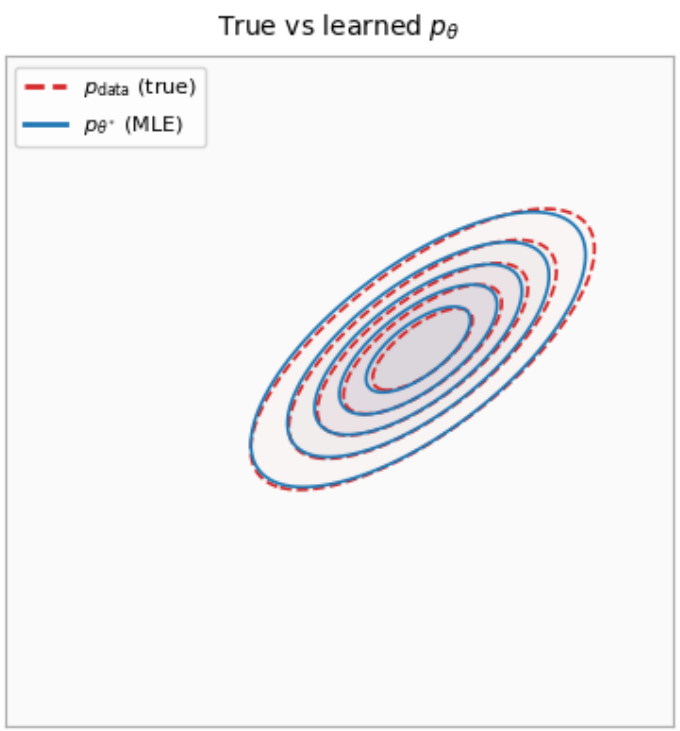
$$\hat{\boldsymbol{\mu}} = \begin{pmatrix} 0.947 \\ 0.497 \end{pmatrix}$$

$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} 1.22 & 0.633 \\ 0.633 & 0.756 \end{pmatrix}$$

$$\boldsymbol{\mu}_{\text{true}} = \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}$$

$$\boldsymbol{\Sigma}_{\text{true}} = \begin{pmatrix} 1.2 & 0.7 \\ 0.7 & 0.8 \end{pmatrix}$$





Example: parametric density model

Choose Gaussian p_θ with $\theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

Closed-form MLE for Gaussian (see Appendix B)

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)}, \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} \left(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}\right)\left(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}\right)^\top$$

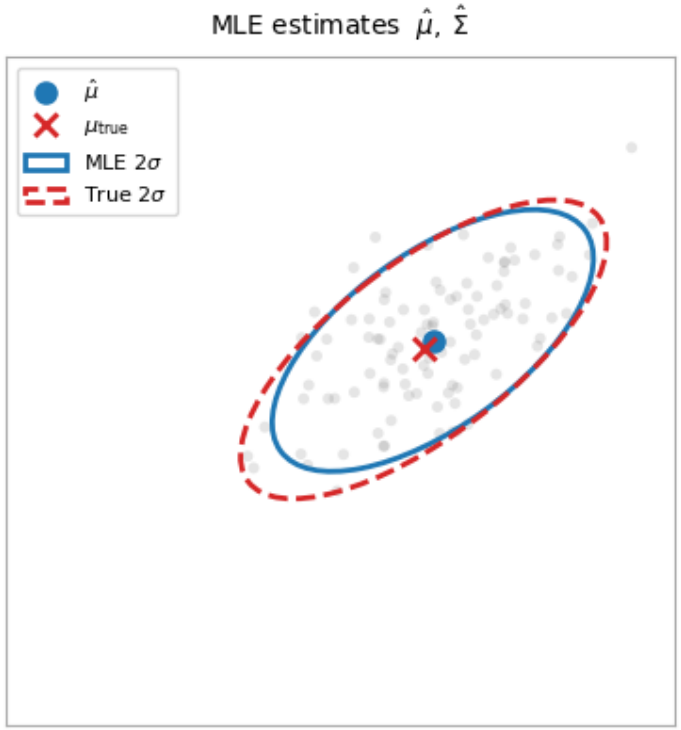
MLE result (100 data points):

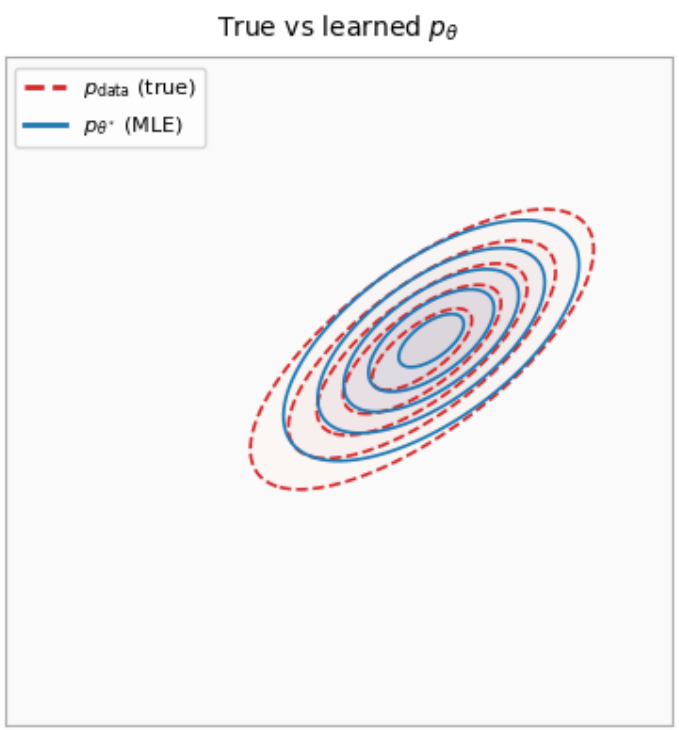
$$\hat{\boldsymbol{\mu}} = \begin{pmatrix} 1.111 \\ 0.606 \end{pmatrix}$$

$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} 0.924 & 0.474 \\ 0.474 & 0.614 \end{pmatrix}$$

$$\boldsymbol{\mu}_{\text{true}} = \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}$$

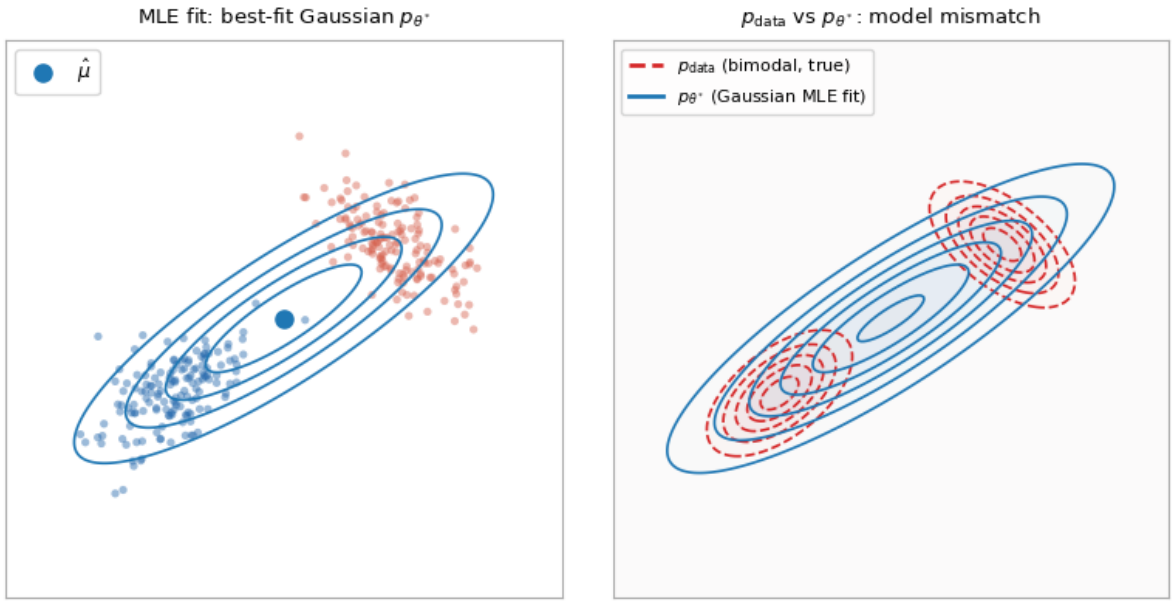
$$\boldsymbol{\Sigma}_{\text{true}} = \begin{pmatrix} 1.2 & 0.7 \\ 0.7 & 0.8 \end{pmatrix}$$





Example: limitations of simple parametric density models

Bimodal data distribution, Gaussian density model



- Classic parametric families may be too restrictive to represent complex data distributions, leading to unavoidable **model bias** even with infinite data.
- Neural networks greatly increase model flexibility through ***universal approximation*** – but direct likelihood optimisation is still challenging.

Why direct KL / MLE optimisation is difficult

With neural network parametrisations p_θ

- A density must satisfy

$$\int p_\theta(\mathbf{x})\, d\mathbf{x} = 1$$

  but enforcing this constraint is difficult.

- Intractable normalisation – numerical integration suffers from the curse of dimensionality; $O\left(M^d\right)$ evaluations

$$\int_{\mathbb{R}^d} p_\theta(\mathbf{x})\, d\mathbf{x} \approx \sum_{j_1=1}^{M} \sum_{j_2=1}^{M} \cdots \sum_{j_d=1}^{M} p_\theta(x_{j_1}, x_{j_2}, \ldots, x_{j_d}) \prod_{k=1}^{d} w_{j_k}$$

- Neural network parameterisations make the KL / MLE objective non-convex, so the global optimum is not guaranteed.

Modern methods such as ***normalising flows***, ***diffusion***, *and* ***VAE*** models address these challenges using different strategies for tractable training and density estimation.

Strategies for tractable density estimation

- **Normalising flow** – parameterise an invertible *change of variables*

$$\mathbf{z} \sim p_0(\mathbf{z}), \qquad \mathbf{x} = f_\theta(\mathbf{z})$$

Total probability mass is 1 by construction

$$\int_{\mathbb{R}^d} p_\theta(\mathbf{x})\, d\mathbf{x} = \int_{\mathbb{R}^d} p_0(f_\theta^{-1}(\mathbf{x}))\left|\det J_{f_\theta^{-1}}(\mathbf{x})\right| d\mathbf{x} = \int_{\mathbb{R}^d} p_0(\mathbf{z})\, d\mathbf{z} = 1$$

- **Diffusion** / **flow matching** avoid explicit density normalisation by learning probability dynamics rather than modelling the density p_θ directly
  - Similar to normalising flows, except the change of variables is replaced by continuous-time dynamics, e.g.

$$\mathbf{z}_t := f_\theta(t, \mathbf{z}_0), \quad \mathbf{z}_0 \sim p_0 \quad \text{and} \quad d\mathbf{z}_t = \mathbf{v}_\theta(t, \mathbf{z}_t)dt$$

  - Sampling requires solving deterministic or stochastic differential equations.

- **Variational auto-encoder** (VAE) – optimise a lower bound on $\log p_\theta\,(\mathbf{x})$ (see Appendix C)

$$\log p_\theta(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))}_{\text{Evidence Lower Bound (ELBO)}}$$

decoder — encoder — prior over latent $\mathbf{z}$

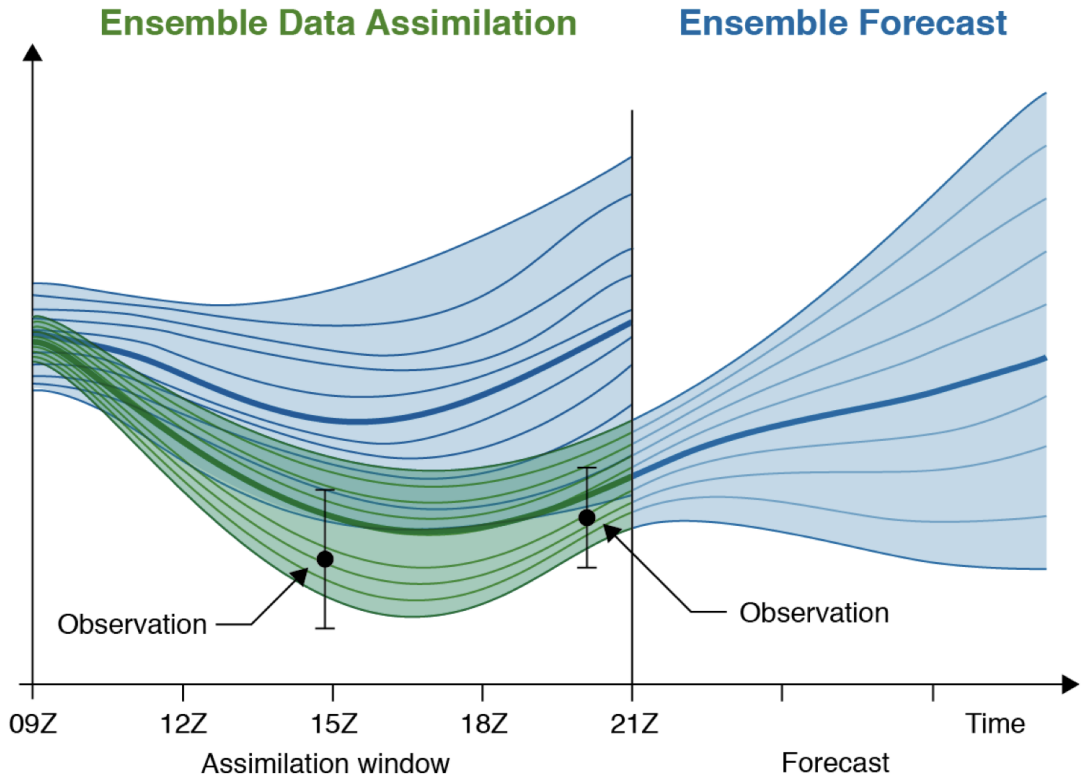
Ensemble 4D-Var

In the 4DVar cost function, the unknown quantities are

- background state $\mathbf{x}_\mathrm{b}$ – apriori estimate of $\mathbf{x}^\mathrm{true}$
- background covar $\mathbf{B}$ – encodes how

$$\delta\mathbf{x}^b := \mathbf{x}^\mathrm{true} - \mathbf{x}^b$$

is correlated spatially and vertically.



The EDA provides flow-dependent estimates of analysis and background error uncertainty (*Isaksen et al* (2010)),

$$\mathbf{B}(t) \approx \frac{1}{N-1}\sum_{i=1}^{N}[\mathbf{x}^{b,(i)}(t) - \bar{\mathbf{x}}^b(t)]\,[\mathbf{x}^{b,(i)}(t) - \bar{\mathbf{x}}^b(t)]^T.$$

Operationally (see *Isaksen et al* (2010), and Eq (1) in *Bonavita et al.* (2015))

$$\mathbf{B}(t) = \mathbf{T}^{-1}\,\boldsymbol{\Sigma}^{\frac{1}{2}}(t)\,\mathbf{C}(t)\,\boldsymbol{\Sigma}^{\frac{1}{2}}(t)\,\mathbf{T}^{-T}$$

balance — correlations — variance

Hybrid 4DVar-ML EDA

Aim – Develop an EDA emulator and integrate it into the operational EDA workflow.

Two complementary directions:

1. EDA statistics
    - Emulate full-EDA statistics using fewer members.

$$\{\mathbf{x}^{b,1}, \mathbf{x}^{b,2}, \ldots, \mathbf{x}^{b,N}\} \longrightarrow \mathbf{B}$$

$$\{\mathbf{x}^{b,1}, \mathbf{x}^{b,2}, \ldots, \mathbf{x}^{b,M}\} \longrightarrow \mathrm{ML} \longrightarrow \tilde{\mathbf{B}}, \quad M < N$$

2. EDA members
    - Generate EDA members that effectively mimic the full EDA.
        - Full 4DVar EDA ensemble $\{\mathbf{x}^{a,1}, \mathbf{x}^{a,2}, \ldots, \mathbf{x}^{a,N}\}$
        - Hybrid 4DVar and ML EDA ensemble $\{\mathbf{x}^{a,1}, \mathbf{x}^{a,2}, \ldots, \mathbf{x}^{a,M}\} \cup \{\tilde{\mathbf{x}}^{a,j_1}, \tilde{\mathbf{x}}^{a,j_2}, \ldots, \tilde{\mathbf{x}}^{a,j_{N-M}}\}$ emulated

EDA variance emulator

See ECMWF Tech Memo 936 for details; doi: 10.21957/0b7e4d4426

- The variance **Σ** part of **B** is just the pointwise unbiased sample variance

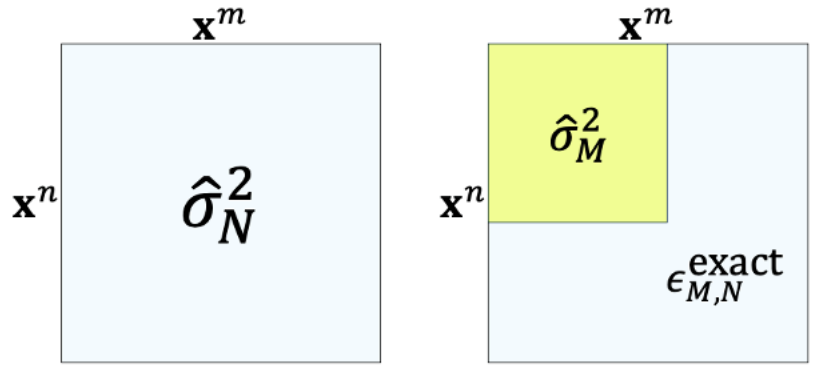
$$\hat{\sigma}_N^2 := \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}^{b,(n)} - \bar{\mathbf{x}}^b)^{\otimes 2} = \frac{1}{2N(N-1)}\sum_{n,m=1}^{N}(\mathbf{x}^{b,(n)} - \mathbf{x}^{b,(m)})^{\otimes 2}$$

- Fix a small ensemble estimate, M < N

where $\alpha_{M,N} = \dfrac{M(M-1)}{N(N-1)}$

$$\hat{\sigma}_N^2 = \alpha_{M,N}\,\hat{\sigma}_M^2 + \epsilon_{M,N}^{\text{exact}},$$

$$\epsilon_{M,N}^{\text{exact}} = \frac{1}{2N(N-1)}\left(2\sum_{n=M+1}^{N}\sum_{m=1}^{M} + \sum_{n,m=M+1}^{N}\right)(\mathbf{x}^n - \mathbf{x}^m)^{\otimes 2}.$$



- The proposed model takes as input $\hat{\sigma}_M$. Its output is a probability distribution on $\hat{\sigma}_N$

$$\hat{\sigma}_M \mapsto \mathbb{P}_\theta(\sigma_N \mid \hat{\sigma}_M).$$

- We use a VAE-based formulation – over the training dataset $(\hat{\sigma}_{M,k}, \hat{\sigma}_{N,k}), \quad k = 1, 2, \ldots, K,$ find parameters that maximise the *Evidence Lower BOund* (ELBO).

$$\theta^* = \arg\max_\theta \mathcal{L}_{\text{ELBO}}(\theta;\, \{\hat{\sigma}_{M,k}, \hat{\sigma}_{M,k}\})$$

EDA variance emulator

We can sample from the learnt distribution,

$$\tilde{\sigma} = g_{\theta_2}\left(f_{\theta_1}(\hat{\sigma}_M), \epsilon\right), \quad \epsilon \sim q_{\theta_3}\left(\cdot \mid \hat{\sigma}_M\right)$$

decoder — encoder — regression error distribution

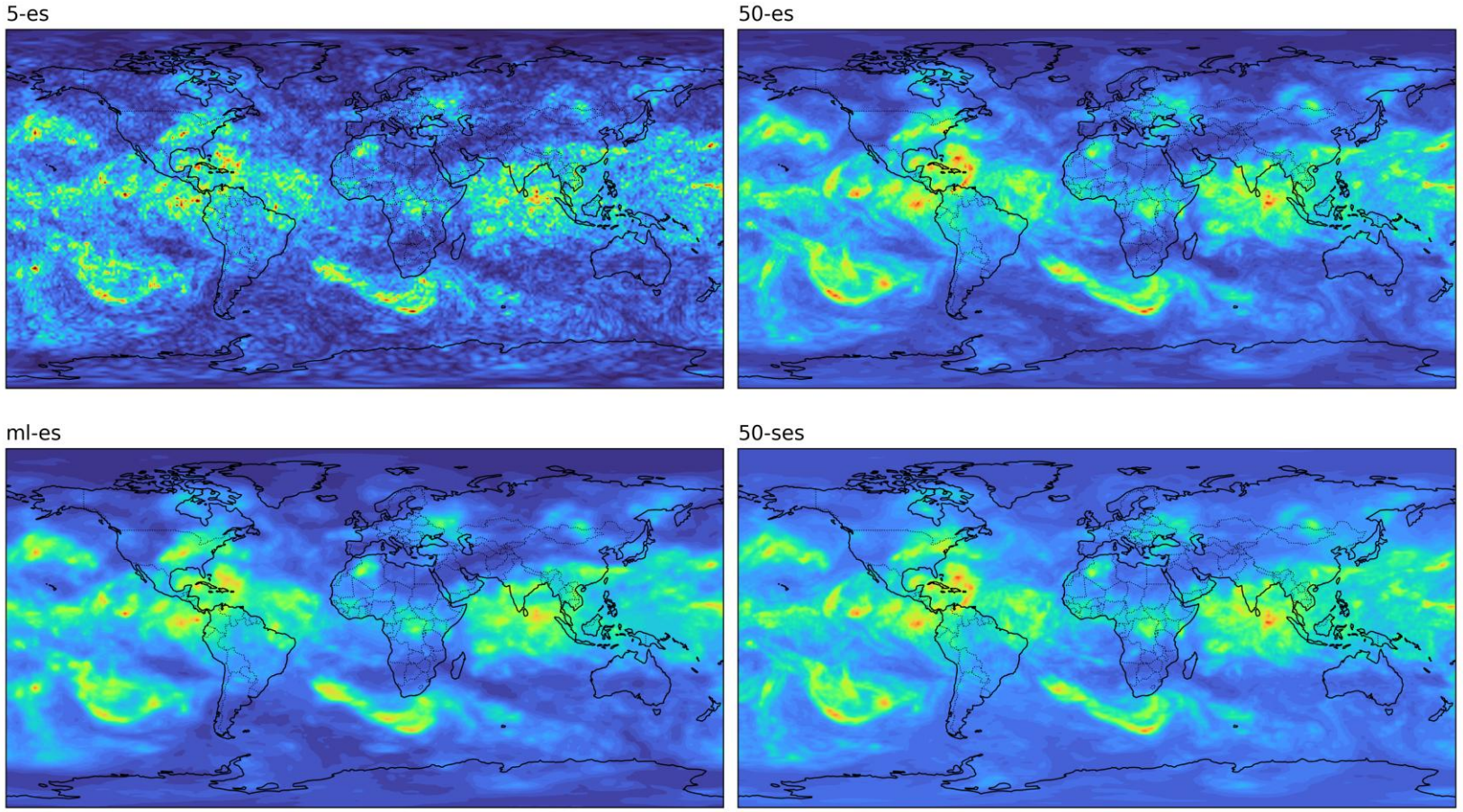
or use the first moment to approximate $\hat{\sigma}_N$.

M=5, N=50
N80 grid, 137 model levels
6 variables: (vo, div, t, lnsp, q, o3)

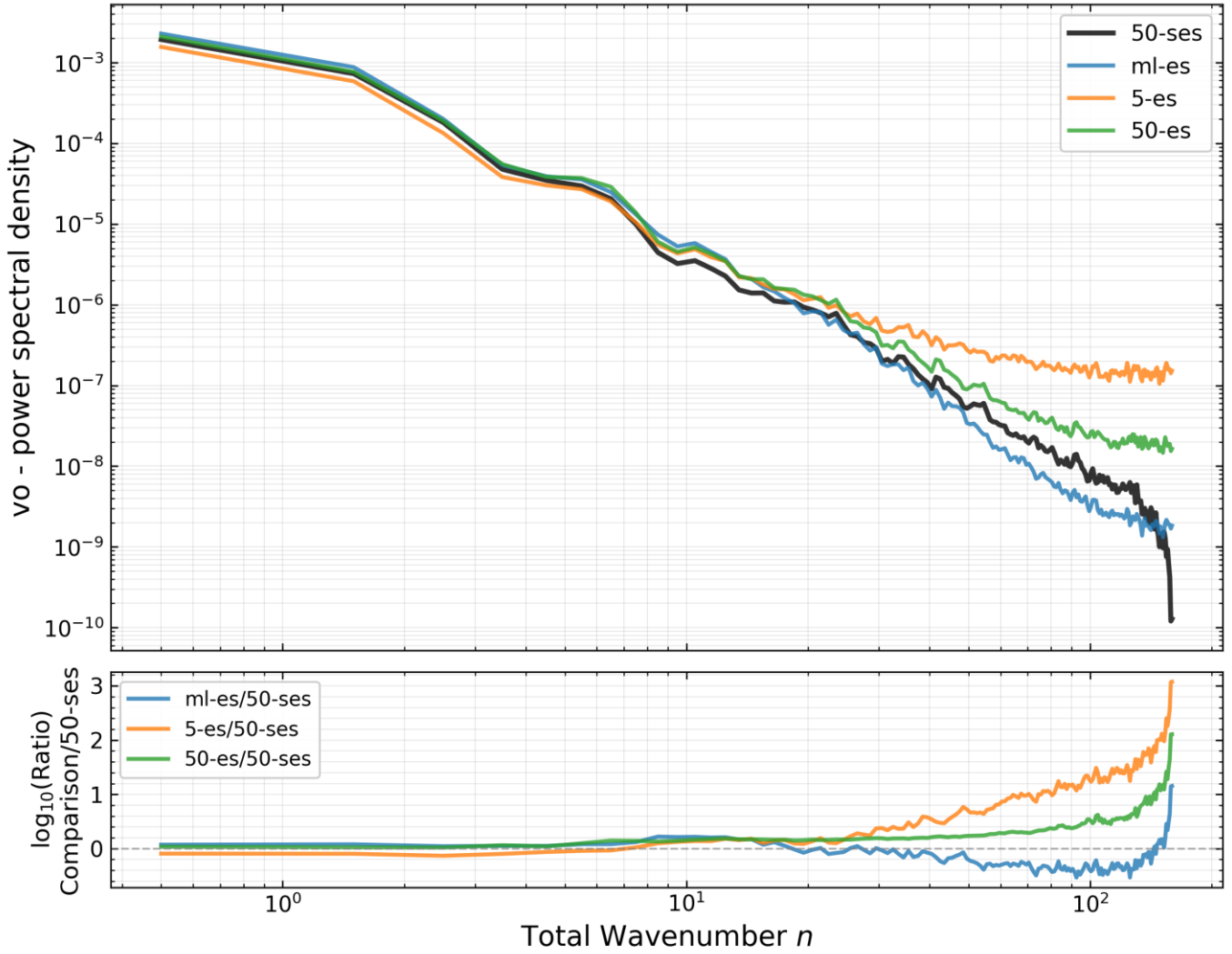
The training dataset spans
01/01/2023 – 31/12/2023.

Each model has
< 200,000 learnable parameters

e.g. Vorticity es, ml74 (~200hPa)
21:00 UTC 01 June 2022



EDA variance estimator



$$\log_{10}\left(\frac{P_X(k)}{P_{\text{50-ses}}(k)}\right),$$

Theoretical constraints

- **Accuracy** – perfect "replication" is not possible due to information theoretic lower bounds;

$$\mathbb{E}\left[(\bar{\sigma} - \hat{\sigma}_N)^2_{i,(\lambda,\phi)}\right] \geq \mathbb{E}\left[(\mathbb{E}(\hat{\sigma}_N|\mathscr{F}_M) - \hat{\sigma}_N)^2_{i,(\lambda,\phi)}\right],$$

the lower bound is strictly positive for $N > M$.

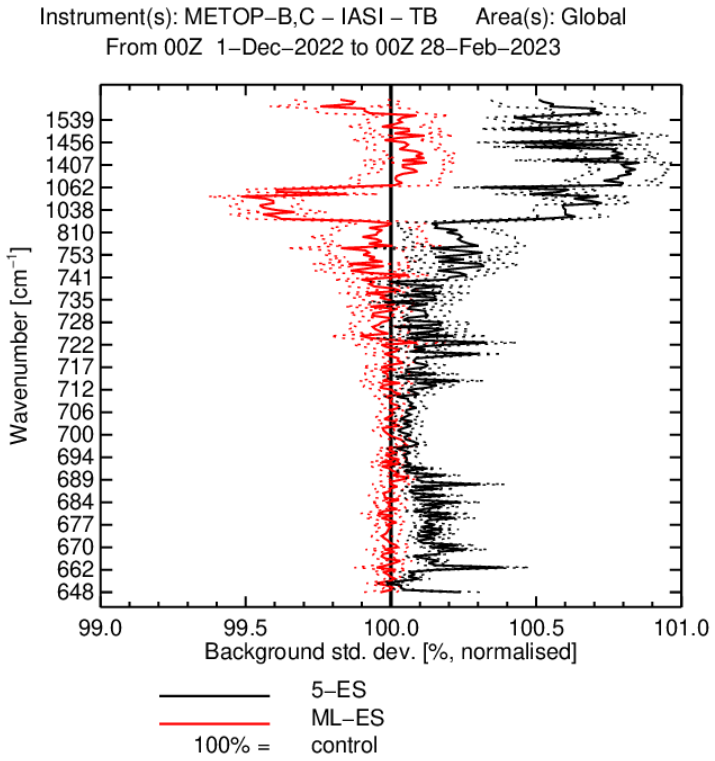
Optimal with respect to mean square error

any estimator from M samples

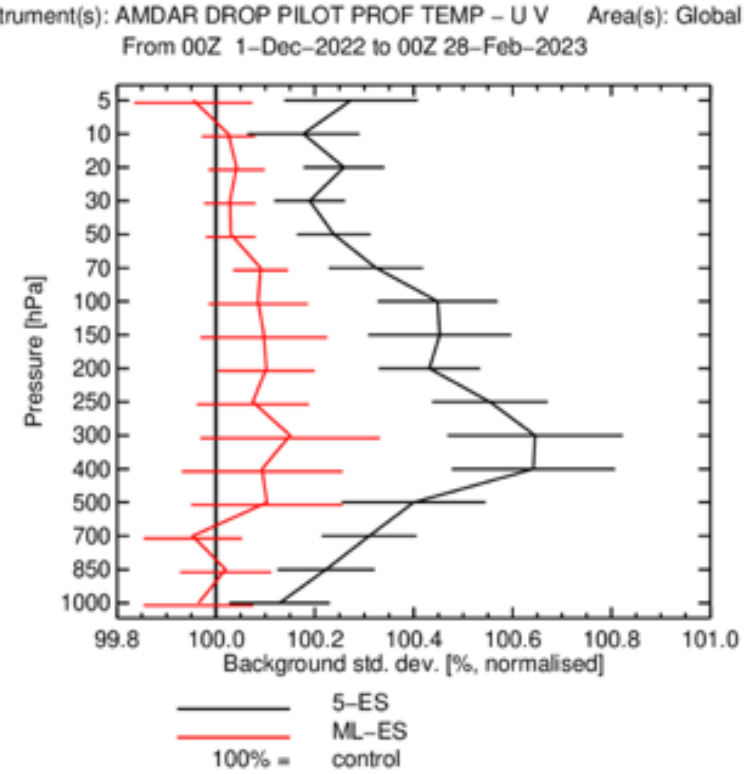
- **Impact on analysis** – (strong-constraint) 4DVar optimisation is ***locally Lipschitz*** **continuous** with respect to **B** (for proof see Tech memo 936 Appendix B), i.e. a small change in **B** means a small change in the resulting analysis increment.

Normalised observation statistics (obstats)

Satellite instruments



Conventional instruments

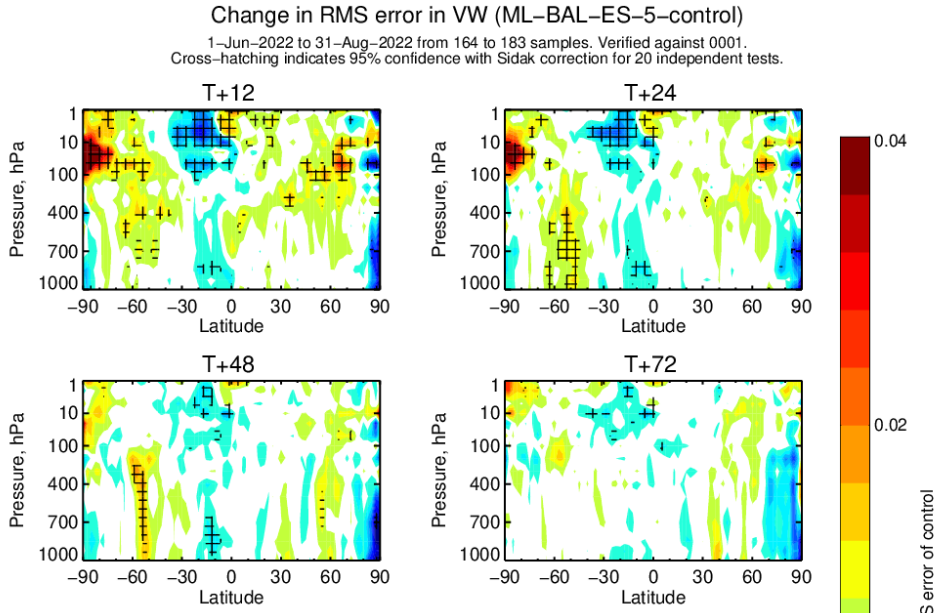


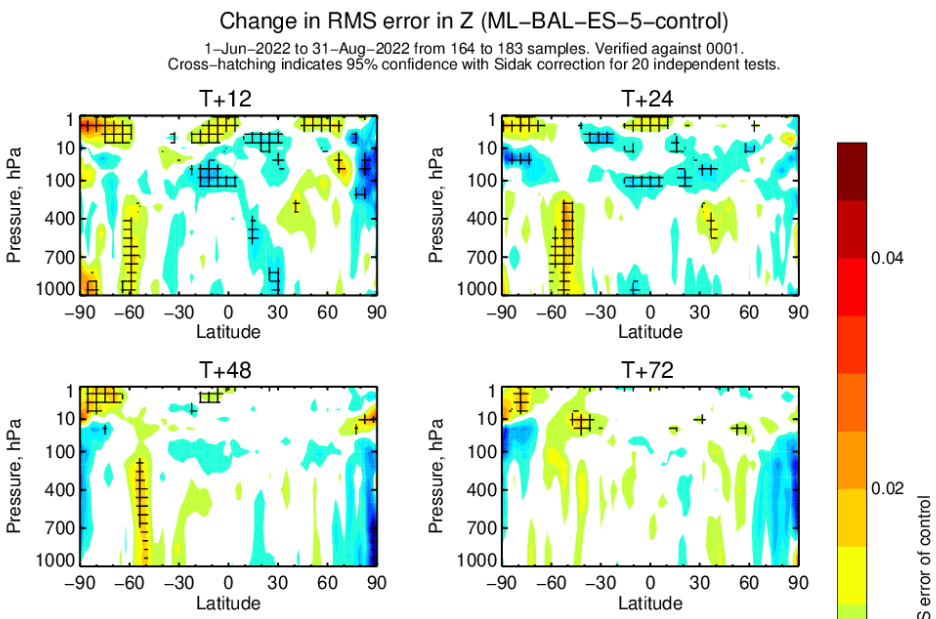
ECMWF

Normalised change in RMS forecast error

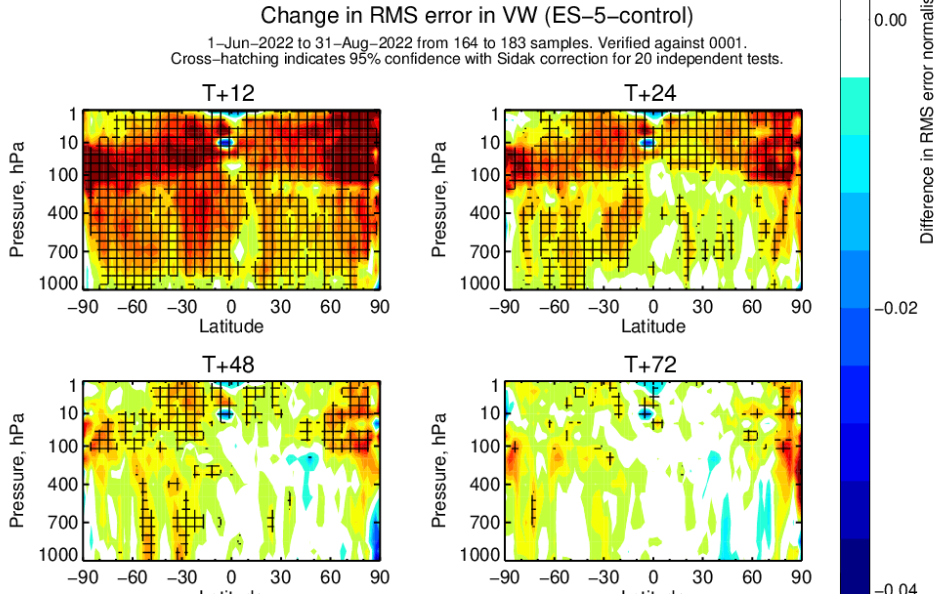
ML-es from 5 members

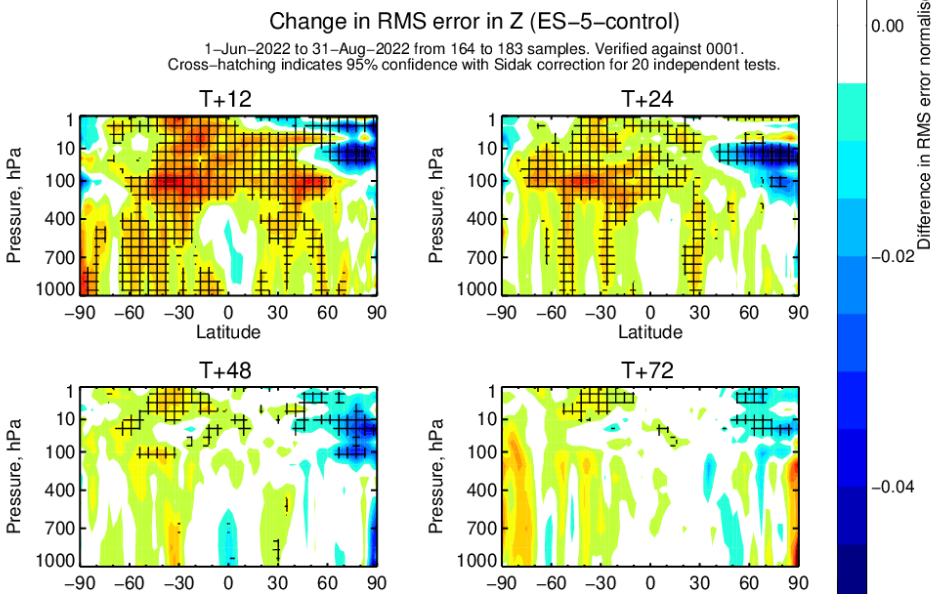
vs 50-es





5-es vs 50-es





EDA Emulation

The EDA system provides empirical background and analysis distributions

$$\mathbb{P}_N^b := \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathbf{x}^{b,(i)}}, \qquad \mathbb{P}_N^a := \frac{1}{N}\sum_{i=1}^{N}\delta_{\phi(\mathbf{x}^{b,(i)}|\mathbf{y}_{\text{obs}})}$$

where ϕ denotes the 4D-Var transformation applied to the background states.

We wish to learn a **hybrid distribution** that is trained to match $\mathbb{P}_N^a$

guidance

$$\mathbb{P}_N^{a,\text{hybrid}} := \frac{1}{N}\left(\sum_{i=1}^{N_s}\delta_{\phi(\mathbf{x}^{b,(k_i)}|\,\mathbf{y}_{\text{obs}})} + \sum_{j=1}^{N-N_s}\delta_{\phi_\theta\left(\mathbf{x}^{b,(k_j)}|\,\mathbb{P}_{N_s}^a\right)}\right)$$

Small 4DVar ensemble distribution $\mathbb{P}_{N_s}^a$

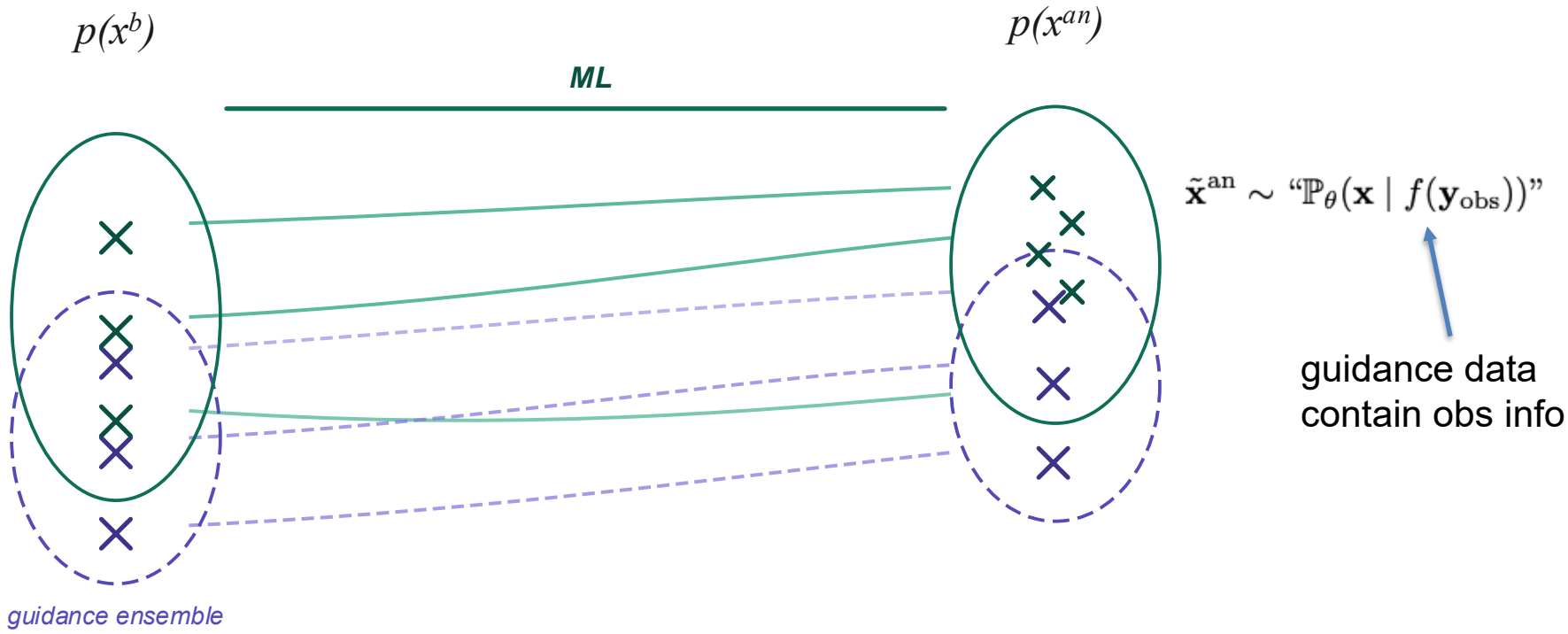
Generative model

The generative model we propose is described by a (deterministic or stochastic) **flow** conditioned on the distribution of the small 4DVar ensemble. Stochasticity enables cheap ensemble enlargement.

$$\phi_t^{(j)} = u_\theta(\phi_t^{(j)}, \mathbb{P}_t, t \mid \mathbb{P}_{N_s}^a)dt + \sigma_\theta(\phi_t^{(j)}, \mathbb{P}_t, t \mid \mathbb{P}_{N_s}^a)\circ dW_t, \quad \phi_0^{(j)} \in \left\{\mathbf{x}^{b,(k_j)}\right\}_{j=1}^{N-N_s}$$

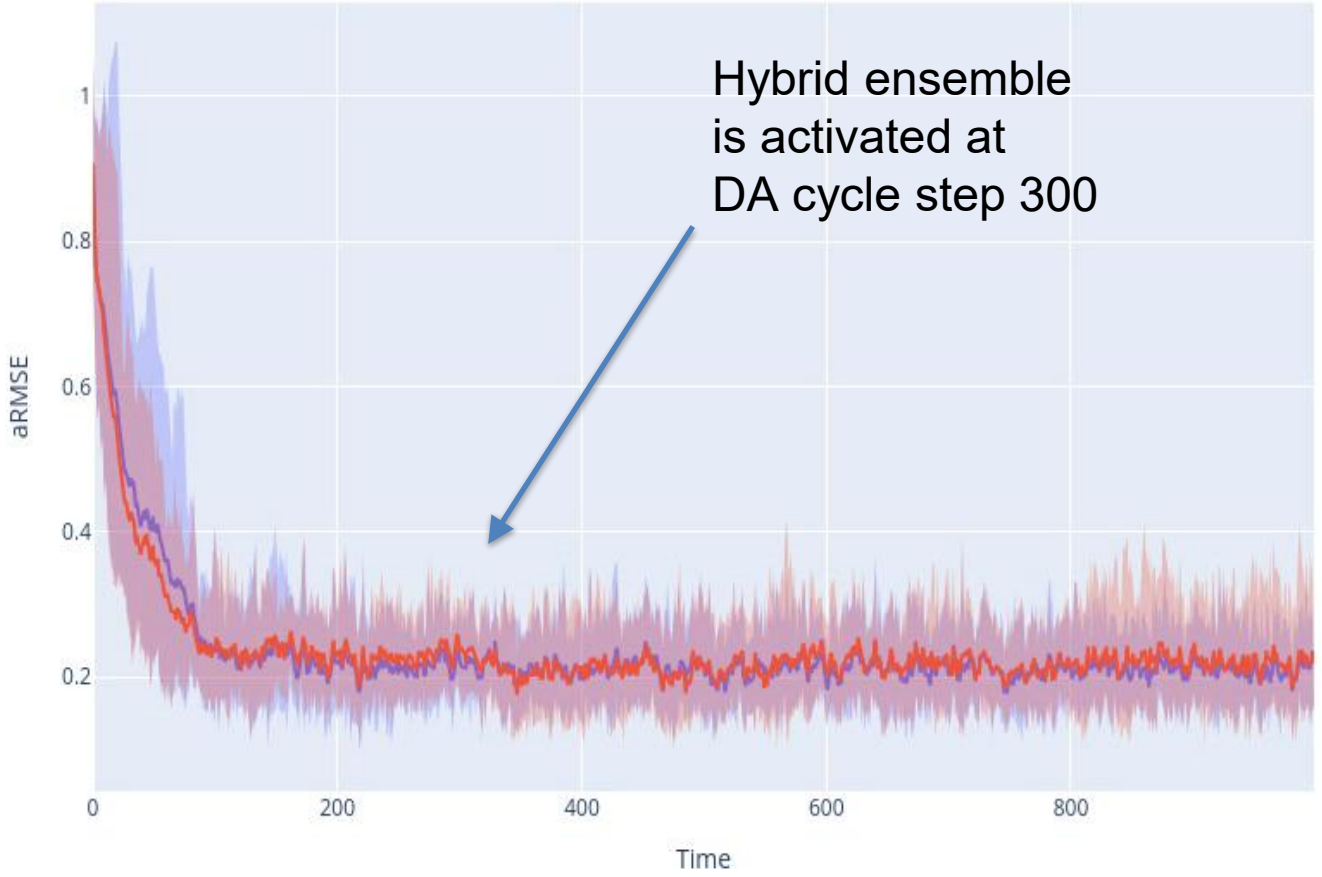
EDA Emulation

Intuitive picture – the small ensemble of 4DVar analyses guides the generative model, so that observations are assimilated indirectly.

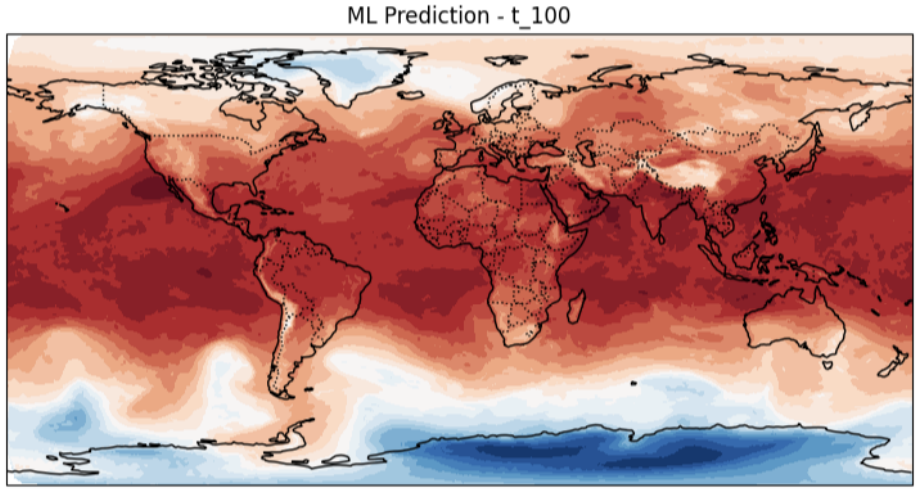


EDA Emulation: toy model cycled EDA experiment

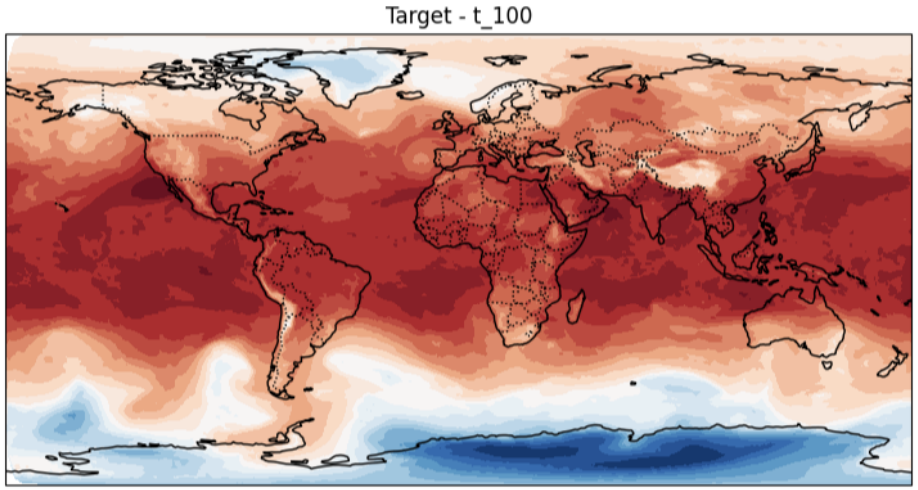
- **Lorenz96 ETKF** (from Alban Farchi)
  - dim=40, N=20 (theoretical minimum ens. size 14)
  - Inflation only, no localisation
- Hybrid ensemble $N_s =$ 10 and 10 emulated.
- Neutral RMSE scores – Hybrid EDA vs Full EDA



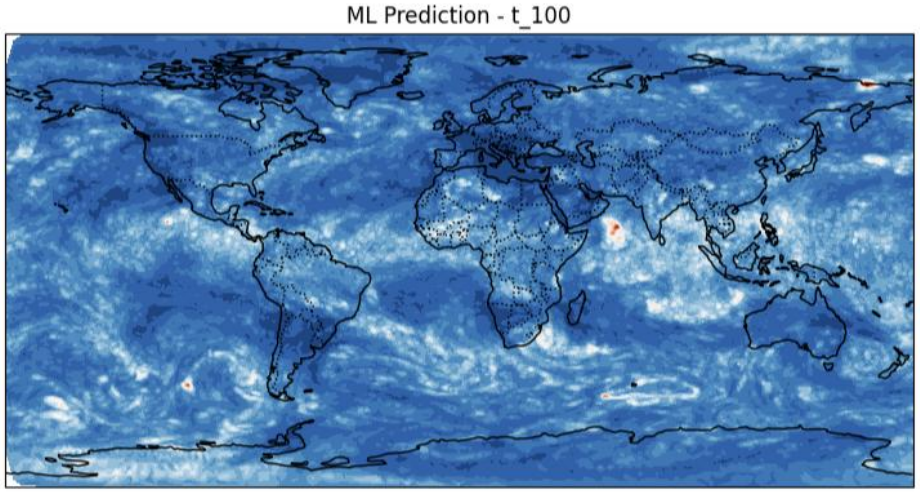
EDA Emulation: Early Experimental Results



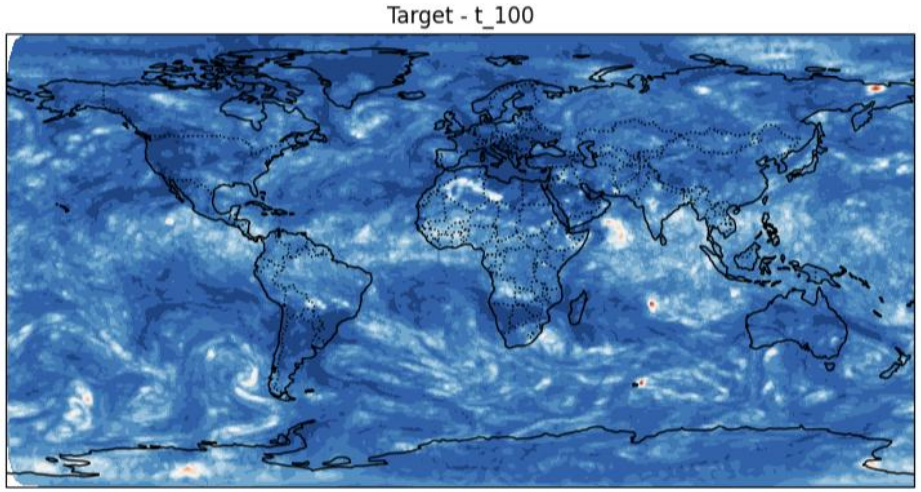
emulated member



EDA member



es (14 EDA + 36 emulated)



es (50 EDA)

ECMWF

Appendix A.1 Intuition on KL divergence / relative entropy

- In information theory, $-\log p(x)$ measures the "**surprise**" of an "event" x that has probability $p(x)$.
- **Entropy:**

$$H(p) = \mathbb{E}_{p(x)}(-\log p(x))$$

is the "average surprise". If you believe $p(x)$, then $H(p)$ is the minimum possible average surprise.

- **Cross entropy:** if you believe $p_\theta(x)$, then the average surprise when the data is actually from $p(x)$ becomes

$$H(p, p_\theta) = \mathbb{E}_{p(x)}(-\log p_\theta(x)).$$

- **Relative entropy:** the extra surprise from believing the wrong model

$$D_{\mathrm{KL}}(\, p \,||\, p_\theta \,) = \mathrm{CrossEntropy} - \mathrm{Entropy}.$$

Homework: *Biased coin, with* $p(H) = 0.9$, $p(T) = 0.1$. *But Alice believes the coin is fair, i.e.* $p_\theta(H) = 0.5 = p_\theta(T)$. *Calculate* $D_{\mathrm{KL}}(\, p \,||\, p_\theta \,)$, *using base* e *logarithm.*

Hint: $\mathbb{E}_{p(x)}(-\log p_\theta(x)) = -p(H)\log p_\theta(H) - p(T)\log p_\theta(T)$

Alice underestimates $p(H)$ – too surprised too often (positive KL contribution).
Alice overestimates $p(T)$ – less surprised than she should be (negative KL contribution).

Evaluate the reverse $D_{KL}(\, p_\theta \,||\, p \,)$. *Do you get the same result as before?*
The distribution used to take the expectation determines which events receive the most weight.

Appendix A.2: Non-negativity of KL divergence

Under the assumption that p_{data} is *absolutely continuous* with respect to p_θ (i.e. if $p_{\text{data}} > 0$, then $p_\theta > 0$), we have

$$-D_{\text{KL}}(p_{\text{data}} \,\|\, p_\theta) = -\int \log \frac{p_{\text{data}}(\mathbf{x})}{p_\theta(\mathbf{x})} p_{\text{data}}(\mathbf{x})\, d\mathbf{x} = \int \log \frac{p_\theta(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} p_{\text{data}}(\mathbf{x})\, d\mathbf{x}$$

Since log is concave, by Jensen's inequality

$$\begin{aligned} -D_{\text{KL}}(p_{\text{data}} \,\|\, p_\theta) &\le \log \int \frac{p_\theta(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} p_{\text{data}}(\mathbf{x})\, d\mathbf{x} \\ &= \log \int p_\theta(\mathbf{x})\, d\mathbf{x} \\ &= \log 1 = 0. \end{aligned}$$

Jensen's inequality:

$\varphi(\mathbb{E}[X]) \ge \mathbb{E}[\varphi(X)], \quad$ for concave φ.

Or

$$D_{\text{KL}}(p_{\text{data}} \,\|\, p_\theta) \ge 0.$$

Appendix B: multivariate Gaussian maximum likelihood estimation (MLE)

Gaussian likelihood over data:

$$p_\theta(\{\mathbf{x}^{(i)}\}) = p(\{\mathbf{x}^{(i)}\} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \frac{1}{(2\pi)|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}^{(i)} - \boldsymbol{\mu})\right)$$

MLE finds the parameters that maximise the likelihood of the observed data

$$\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}} = \arg\max_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

and

$$l(\boldsymbol{\mu}, \boldsymbol{\Sigma}) := \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{N}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) + \text{const.}$$

Setting $\frac{\partial l}{\partial \boldsymbol{\mu}} = 0$ and $\frac{\partial l}{\partial \boldsymbol{\Sigma}} = 0$

$$\frac{\partial \ell}{\partial \boldsymbol{\mu}} = \sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1}(\mathbf{x}^{(i)} - \boldsymbol{\mu}) = 0$$

$$\frac{\partial \ell}{\partial \boldsymbol{\Sigma}} = -\frac{N}{2}\boldsymbol{\Sigma}^{-1} + \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1}(\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} = 0$$

Calculus identities

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} \log |\boldsymbol{\Sigma}| = \boldsymbol{\Sigma}^{-1}$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} \mathbf{a}^\top \boldsymbol{\Sigma}^{-1} \mathbf{a} = -\boldsymbol{\Sigma}^{-1} \mathbf{a}\mathbf{a}^\top \boldsymbol{\Sigma}^{-1}$$

$$\frac{\partial}{\partial \boldsymbol{\mu}} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = -2\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$$

Rearrange to get the closed-form Gaussian MLE formulas

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)}, \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}\right)\left(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}\right)^\top.$$

Appendix C: Evidence Lower Bound (ELBO)

We wish to maximise the log-likelihood of the observed data (a.k.a. *evidence* in Bayesian statistics),

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})\, d\mathbf{z}$$

where $\mathbf{z}$ denotes the latent variable – think "compressed representation".

This integral is intractable in high dimensions, so we introduce a variational posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$.

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})\, d\mathbf{z} \\
&= \log \int q_\phi(\mathbf{z} \mid \mathbf{x}) \frac{p_\theta(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\, d\mathbf{z} \quad \text{(This is precisely importance sampling)} \\
&= \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\frac{p_\theta(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right] \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{z}) - \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z})} \right] \quad \text{(Jensen's inequality, see Appendix A)}
\end{aligned}$$

gives KL Divergence

This lower bound is the ELBO.

*Optimising ELBO requires a reparametrisation (Kingma & Welling (2014)) of the importance distribution q_ϕ

$$\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x})\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

to isolate randomness from NN parameters, because the sampling operation $\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})$ is not differentiable.

References

EDA

- Isaksen, L., Bonavita, M., Buizza, R., Fisher, M., Haseler, J., Leutbecher, M., & Raynaud, L. (2010). *Ensemble of data assimilations at ECMWF*. ECMWF. https://doi.org/10.21957/OBKE4K60
- Bonavita, M., Hólm, E., Isaksen, L., & Fisher, M. (2015). The evolution of the ECMWF hybrid data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 142(694), 287–303. https://doi.org/10.1002/qj.2652
- Pan, W., Bonavita, M., Chrust, M., & Hólm, E. (2026). *Data-driven emulation of background-error variance in variational data assimilation*. ECMWF. https://doi.org/10.21957/0B7E4D4426

ML (normalising flow, diffusion, VAE)

- Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2017). Density estimation using Real-NVP. *International Conference on Learning Representations (ICLR)*. arXiv:1605.08803
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 37, 2256–2265.
- Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*.