# Computational Optimizations and Emulation of EDA Perturbed Members and Statistics

## ECMWF Workshop on Data Assimilation
## Bonn 9-10 April 2025

**Elas Holm and Wei Pan**

with Jorge Bandeiras, Massimo Bonavita, Marcin Chrust, Alan Geer, Paddy Gillies, Simon Lang, Peter Lean and more contributing ideas and optimizations

ECMWF

© ECMWF 09/04/2025

# Recent Efficiency Gains in EDA (x5) and How Used to Improve EDA

- Recent improvements in the computational efficiency of the EDA from technical and algorithmic developments (factor 5 since 2019) has been used to increase the consistency and accuracy of the EDA using same or reduced fraction of the ECMWF supercomputers.
- The two main applications of EDA are to provide background errors B for the analysis (deterministic and ensemble) and contribute perturbations for ENS initial conditions. The EDA developments have at each stage **improved deterministic and ensemble forecast scores**.

# Recent Efficiency Gains in EDA (x5) and How Used to Improve EDA

- **2019 46R1:** 50% technical efficiency gain enables EDA members 25-->50 at same cost to match 50 ENS members number (both at 18km) ==>
  - improved high-resolution analysis (better background error B),
  - Improved EDA-based perturbations for the IC's of 50-member ENS,
  - Exchangeability of members improves validity of RD tests with few members ENS.

- **2024 49R1:** 30% algorithmic efficiency gain from soft re-centring enables EDA 18km-->9km to match ENS resolution (meanwhile increased to 9km), while using same proportion of new ATOS clusters as the old CRAY clusters. Combined with scientific improvement of model uncertainty parameterization (SPPT-->SPP) ==>
  - **i**mproved spread, improved high-resolution analysis (better background error B),
  - improved ENS initial conditions.

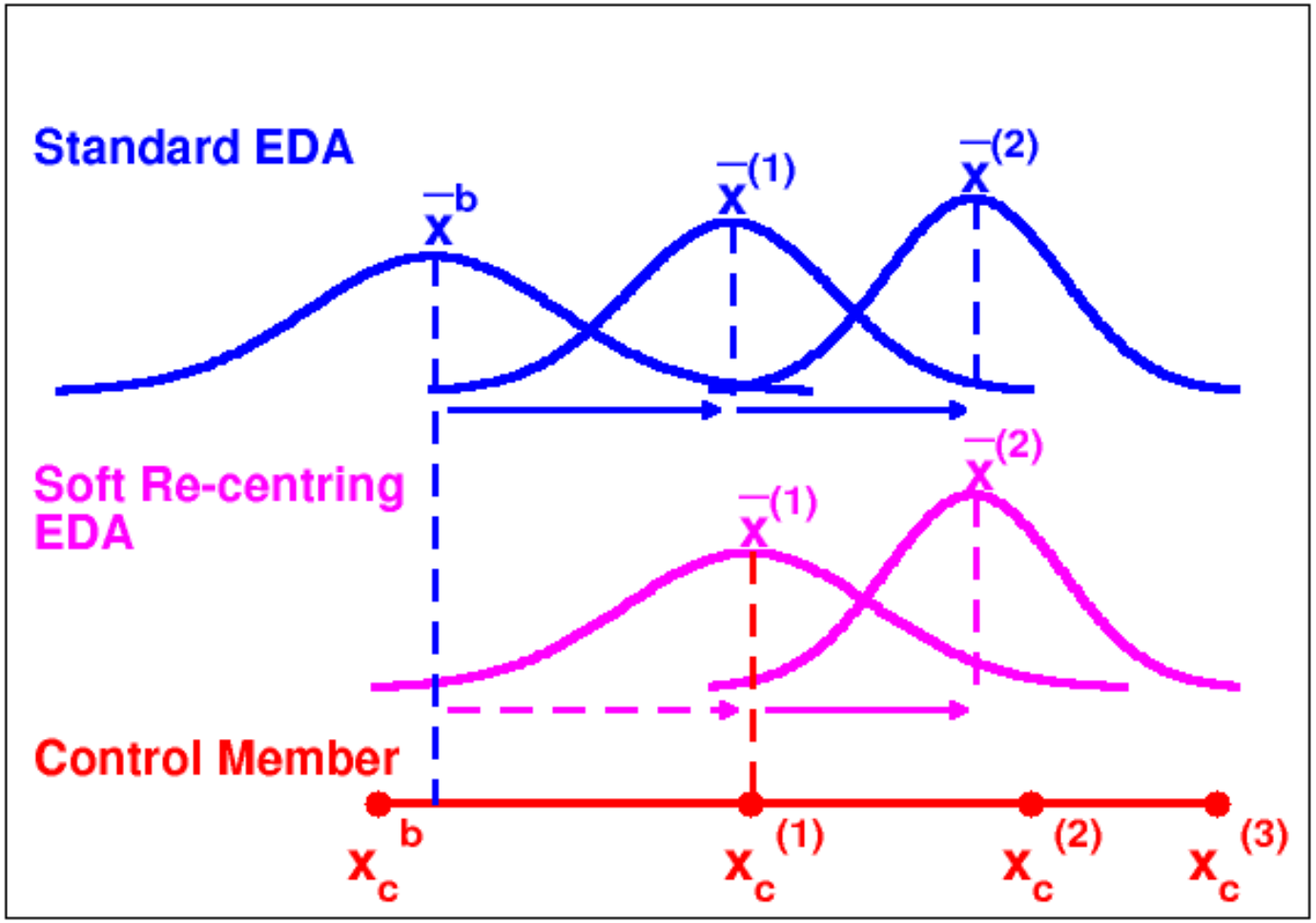
- **2025/6 50R1:** 30-40% technical and algorithmic efficiency gain (single precision trajectory, memory optimize 4D-Var, optimized inner loop resolution) enables to run EDA on 30-40% less resources (memory x wallclock). Combined with scientific improvement of spread calculation ==>
  - **i**mproved high-resolution analysis (better background error B),
  - improved ENS initial conditions.

# 49R1 Soft Re-Centred EDA => One Outer Loop Equals Two!

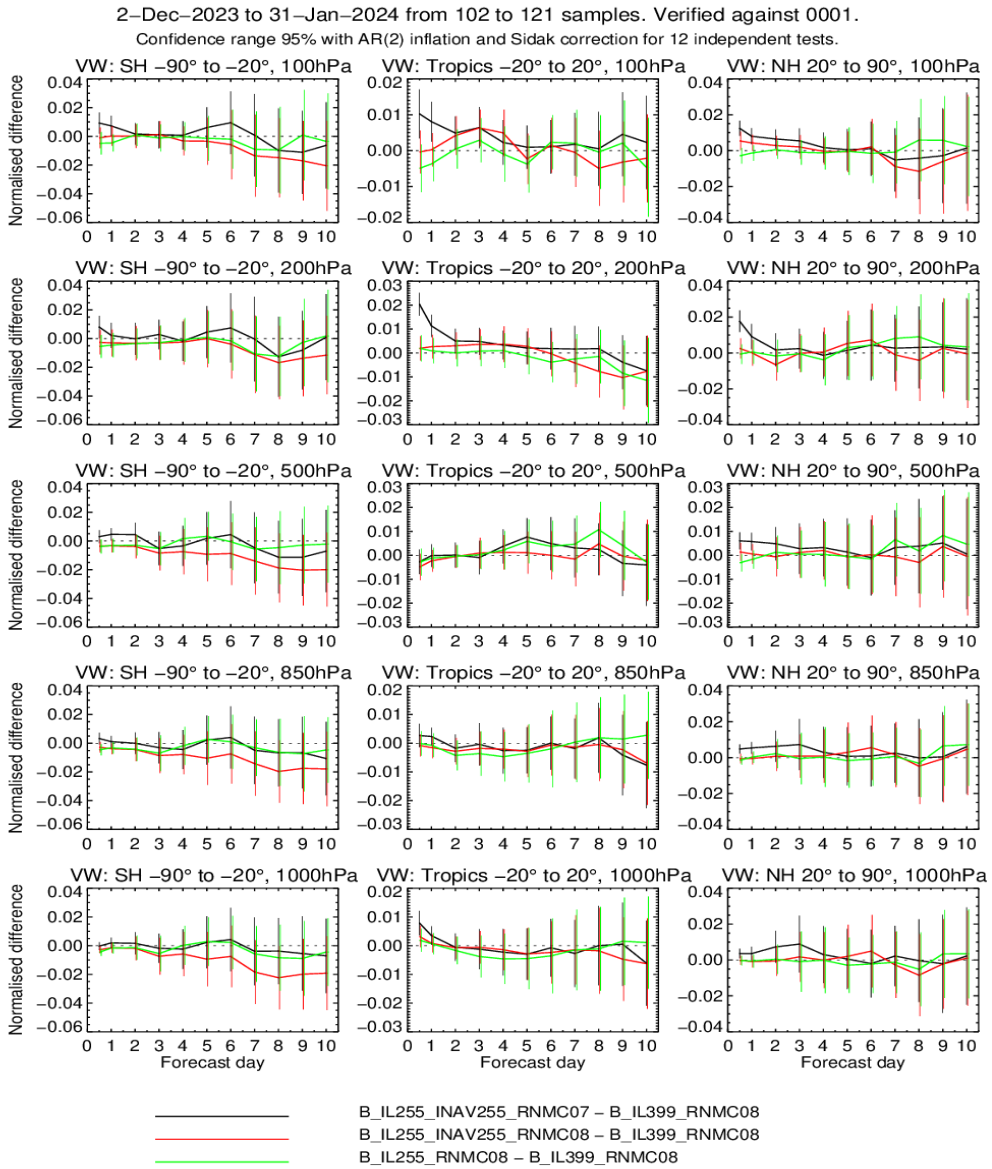
The 49R1 EDA resolution increase to TCo1279 and introduction of SPP was needed for more realistic spread.

Required computational efficiency gain ➔ Soft Re-centred EDA:

- Technically AND quality wise equivalent to doing two outer loops.
- Saves one EDA minimization, 30%, for same quality, enabling higher resolution EDA.
- Control runs more/higher resolution minimization for higher quality.

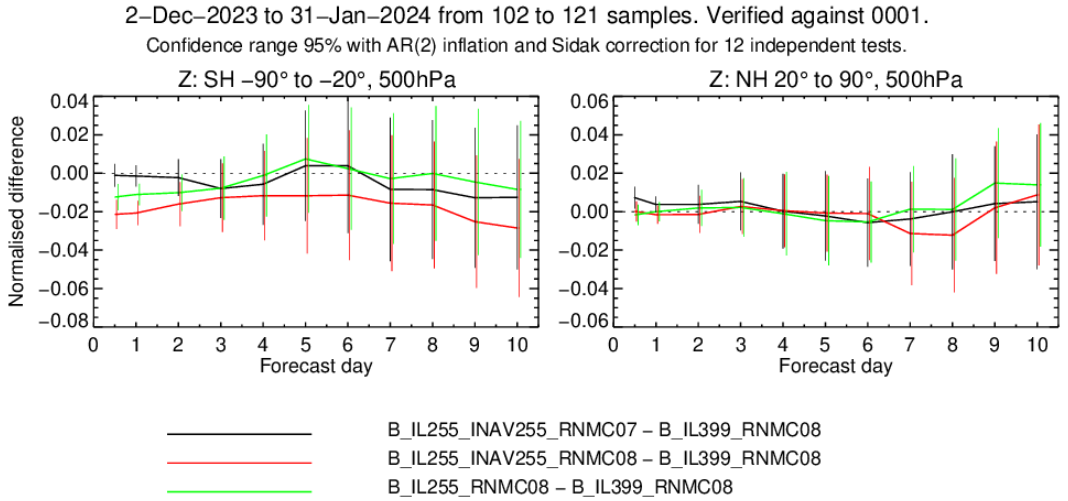


# 50R1 TCo1279 4D-Var using B from 10-mem TCo1279/TL255-319-511 EDA's:




- 4D-Var scores wrt OPER neutral to advantage for TL255 inner loop with T255 spread (red line).
- Only TL255 inner loop not as good (green line).
- REDMNC=0.7 worse (black line), so current 0.8 fine.

- Note: Show OPER scores, OWN rmse scores biased in short range due to spread change of EDA/**B**.




**Wind dRMSE**



**Z500 dRMSE**

# 50R1 TCo1279 10-mem ENS/EDA with 50R1 Re-centring on e-suite+standard SV:

## TL255 IL+T255 Spread vs Reference TL399 IL+T159 spread (Dec-Jan 2023-24)

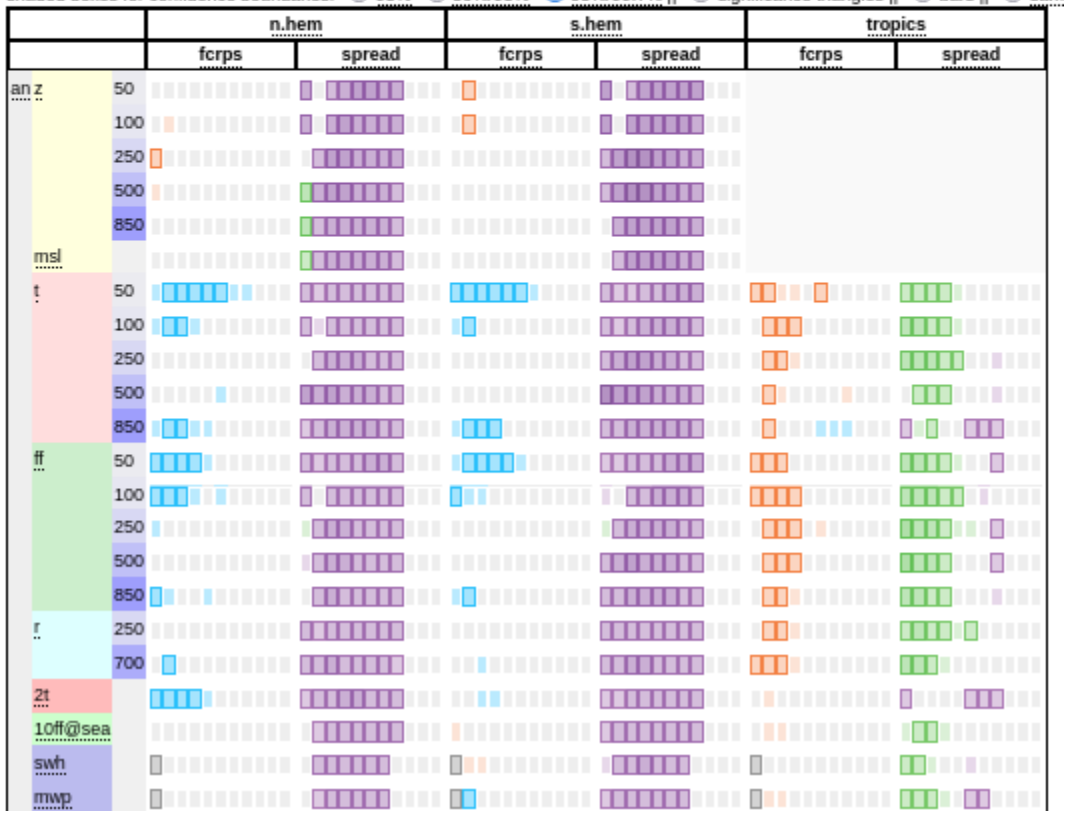
**IL255_INAV255_RMC0.8_vs_IL399_INAV159_RMC0.8 scorecard**

dates=[2023-12-02 00:00:00,2023-12-02 12:00:00,2023-12-03 00:00:00,...,2024-01-31 00:00:00,2024-01-31 12:00:00]
steps=[0, 24, 48, 72, 96, 120, 144, 168, 192, 216, 240]
vstreams=['qrdx_an', 'qrdx_ob']
classs=rd
streams=['enfo', 'waef']
expvers=(cntrl:ik97, exper:ik9a)
reftypes=['an', 'ob']

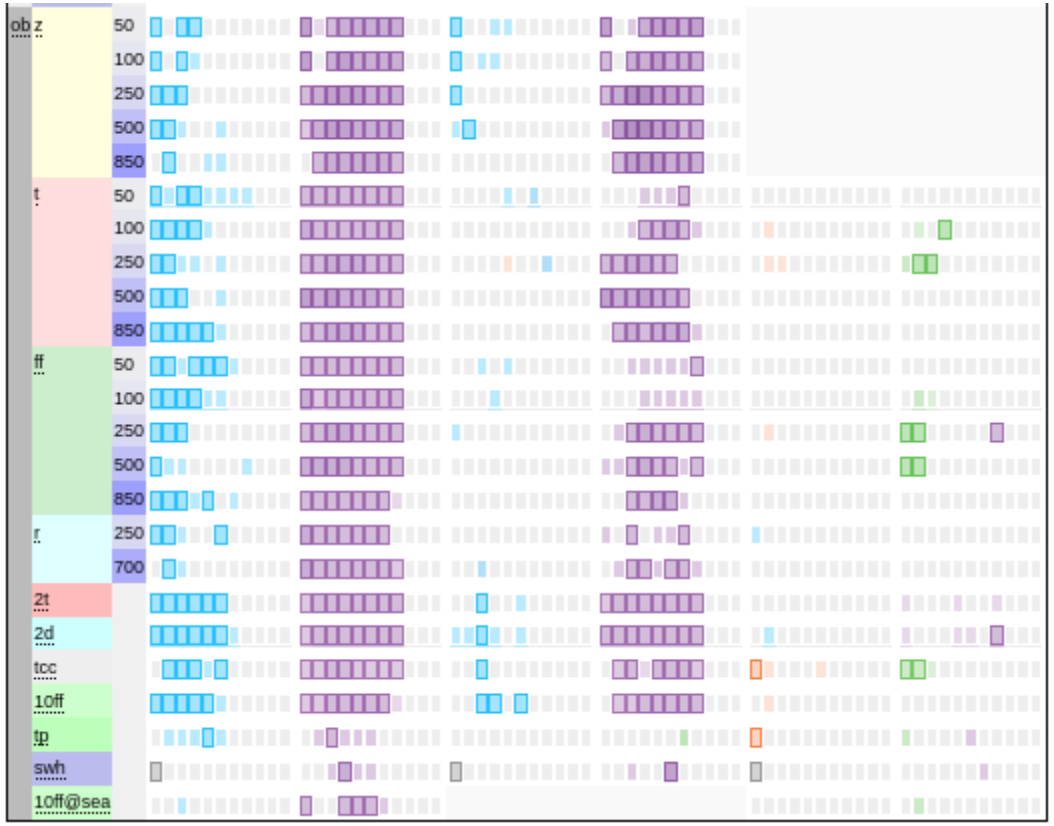
☑ fcrps ☑ spread

☑ n.hem ☑ s.hem ☑ tropics ☐ europe ☐ n.atl ☐ n.pac ( ☑ all)

shaded boxes for confidence boundaries: ○ 95% ○ 50%/95% ◉ 95%/99.7% || ○ significance triangles || ○ bars || ○ sam



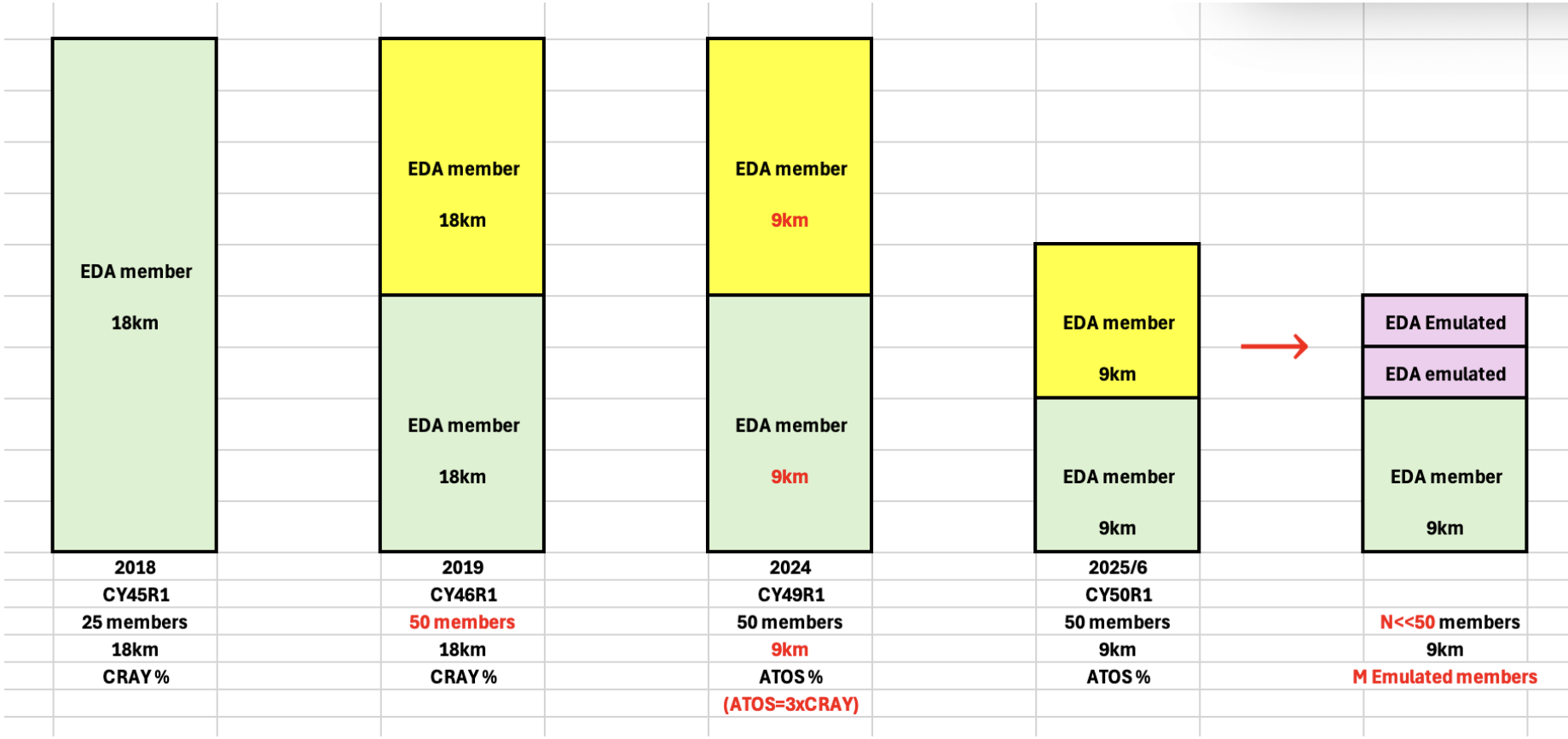

FCRPS wrt OBS/OPER improved/neutral in Extratropics.
Tropics worse short range wrt OPER, neutral wrt OBS
(same as in T255 IL only)




**OPER** **OBS**

# Future Efficiency Gains of EDA: Emulate Subset Members with Fast ML?

- Can we keep only a few full-resolution physical EDA members, while emulating the rest with ML and keep equal or better background errors and ENS initial conditions that improve deterministic and ensemble forecast scores?



# 4DVar Data Assimilation

◇ Bayes theorem + Gaussianity assumption:

$$\underbrace{\pi(\mathbf{x}|\mathbf{y}_{\text{obs}})}_{\textit{posterior}} \propto \underbrace{\exp(-0.5(\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{B}^{-1}(\mathbf{x}-\bar{\mathbf{x}}))}_{\textit{prior}} \underbrace{\pi(\mathbf{y}_{\text{obs}}|\mathbf{x})}_{\textit{likelihood}}. \tag{1}$$

◇ An *analysis*

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \pi(\mathbf{x} \,|\, \mathbf{y}_{\text{obs}}) \tag{2}$$

is the optimiser of

$$L(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{B}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) + \textit{"log of likelihood"} \tag{3}$$

◇ We use *ensemble statistics* to estimate $\bar{\mathbf{x}}$ and $\mathbf{B}$:

$$\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^{50}.$$

ECMWF

© ECMWF 10 Apr 2025

# Cost saving using emulation

**Why** – Optimisation is expensive! Need many evaluations of TL/AD operators.
Can we make things cheaper using alternative methods?

# Cost saving using emulation

**Why** – Optimisation is expensive! Need many evaluations of TL/AD operators.
Can we make things cheaper using alternative methods?

**How** – Build a model to *generate* physically consistent perturbations cheaply.

# Cost saving using emulation

**Why** – Optimisation is expensive! Need many evaluations of TL/AD operators.
Can we make things cheaper using alternative methods?

**How** – Build a model to *generate* physically consistent perturbations cheaply.

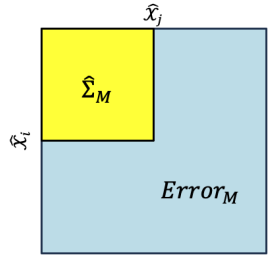
**What**

- Statistics emulator for the 'diagonal' part of $\mathbf{B}$ is being tested for its impact on operational ensemble 4DVar.
- Perturbations generator is in development for generating analysis perturbations.

# Statistics emulator

The 'diagonal' part of $\mathbf{B}$ is denoted by $\mathbf{\Sigma}$, whose entries are pointwise bg error stddev, estimated using EDA sample variance

$$\mathbf{\Sigma}^2 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j>i} (\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j)^{\otimes 2}, \qquad (N = 50 \text{ currently}). \tag{4}$$



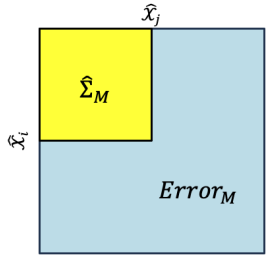
# Statistics emulator

The 'diagonal' part of $\mathbf{B}$ is denoted by $\mathbf{\Sigma}$, whose entries are pointwise bg error stddev, estimated using EDA sample variance

$$\mathbf{\Sigma}^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j>i}(\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j)^{\otimes 2}, \qquad (N = 50 \text{ currently}). \tag{4}$$

We take a subset of EDA members, $\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^M$, $M < N$, calculate

$$\hat{\mathbf{\Sigma}}_M^2 = \frac{1}{M^2}\sum_{i}^{M}\sum_{j>i}^{M}(\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j)^{\otimes 2}, \quad (\text{currently } M = 5) \tag{5}$$



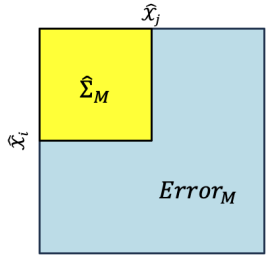
# Statistics emulator

The 'diagonal' part of $\mathbf{B}$ is denoted by $\mathbf{\Sigma}$, whose entries are pointwise bg error stddev, estimated using EDA sample variance

$$\mathbf{\Sigma}^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j>i}(\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j)^{\otimes 2}, \qquad (N = 50 \text{ currently}). \tag{4}$$

We take a subset of EDA members, $\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^M$, $M < N$, calculate

$$\hat{\mathbf{\Sigma}}_M^2 = \frac{1}{M^2}\sum_{i}^{M}\sum_{j>i}^{M}(\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^j)^{\otimes 2}, \quad (\text{currently } M = 5) \tag{5}$$




then have a trained parametric model provide a guess of $\mathbf{\Sigma}$ based on $\hat{\mathbf{\Sigma}}_M$

$$\hat{\mathbf{\Sigma}}_M \longmapsto \mathbf{\Sigma}.$$

Note, by C.L.T.: $\hat{\mathbf{\Sigma}}_M^2 - \mathbf{\Sigma}^2 \sim \frac{1}{\sqrt{M}}\text{sqrt}(\mu_4 - \mathbf{\Sigma}^4)\,\mathcal{N}(0,1)$

# Statistics emulator

Our emulation model is a parametric conditional probability distribution,

$$\hat{\boldsymbol{\Sigma}}_M \longmapsto \mathrm{p}_\theta(\tilde{\boldsymbol{\Sigma}} \,|\, \hat{\boldsymbol{\Sigma}}_M) \quad \xrightarrow{\text{sample}} \tilde{\bar{\boldsymbol{\Sigma}}} \approx \boldsymbol{\Sigma}. \tag{6}$$

# Statistics emulator

Our emulation model is a parametric conditional probability distribution,

$$\hat{\mathbf{\Sigma}}_M \longmapsto \mathrm{p}_\theta(\tilde{\mathbf{\Sigma}} \,|\, \hat{\mathbf{\Sigma}}_M) \quad \xrightarrow{\text{sample}} \tilde{\bar{\mathbf{\Sigma}}} \approx \mathbf{\Sigma}. \tag{6}$$

We implement $\mathrm{p}_\theta(\tilde{\mathbf{\Sigma}} \,|\, \hat{\mathbf{\Sigma}}_M)$ as a nonlinear regression,

$$g_{\theta_1}\left(f_{\theta_2}(\hat{\mathbf{\Sigma}}_M), e\right) \xrightarrow{\text{sample}} \tilde{\mathbf{\Sigma}}, \quad e \sim \mathrm{q}_{\theta_3}(\mathbf{z}|\hat{\mathbf{\Sigma}}_M). \tag{7}$$

# Statistics emulator

Our emulation model is a parametric conditional probability distribution,

$$\hat{\boldsymbol{\Sigma}}_M \longmapsto \mathrm{p}_\theta(\tilde{\boldsymbol{\Sigma}} \,|\, \hat{\boldsymbol{\Sigma}}_M) \quad \xrightarrow{\text{sample}} \tilde{\bar{\boldsymbol{\Sigma}}} \approx \boldsymbol{\Sigma}. \tag{6}$$

We implement $\mathrm{p}_\theta(\tilde{\boldsymbol{\Sigma}} \,|\, \hat{\boldsymbol{\Sigma}}_M)$ as a nonlinear regression,

$$g_{\theta_1}\left(f_{\theta_2}(\hat{\boldsymbol{\Sigma}}_M), e\right) \xrightarrow{\text{sample}} \tilde{\boldsymbol{\Sigma}}, \quad e \sim \mathrm{q}_{\theta_3}(\mathbf{z}|\hat{\boldsymbol{\Sigma}}_M). \tag{7}$$

We build six emulators, one for each model variable (vo, dv, lnsp, q, O3 and t).

Each model has $\sim 199114$ trainable parameters.

# Statistics emulator

Our emulation model is a parametric conditional probability distribution,

$$\hat{\mathbf{\Sigma}}_M \longmapsto \mathrm{p}_\theta(\tilde{\mathbf{\Sigma}} \,|\, \hat{\mathbf{\Sigma}}_M) \quad \xrightarrow{\text{sample}} \tilde{\bar{\mathbf{\Sigma}}} \approx \mathbf{\Sigma}. \tag{6}$$

We implement $\mathrm{p}_\theta(\tilde{\mathbf{\Sigma}} \,|\, \hat{\mathbf{\Sigma}}_M)$ as a nonlinear regression,

$$g_{\theta_1}\left(f_{\theta_2}(\hat{\mathbf{\Sigma}}_M), e\right) \xrightarrow{\text{sample}} \tilde{\mathbf{\Sigma}}, \quad e \sim \mathrm{q}_{\theta_3}(\mathbf{z}|\hat{\mathbf{\Sigma}}_M). \tag{7}$$

We build six emulators, one for each model variable (vo, dv, lnsp, q, O3 and t).

Each model has $\sim 199114$ trainable parameters.

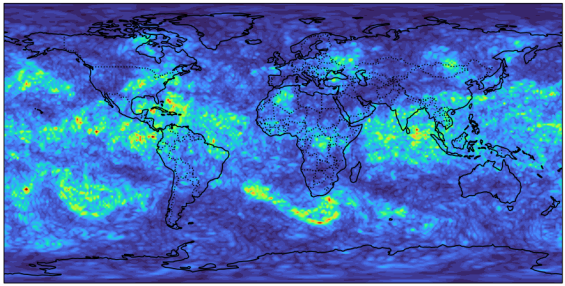
Impact on analysis: consider $\tilde{\mathbf{B}} := \mathbf{B} + \epsilon\delta\mathbf{B}$, (for 3DVar) we have

$$\|d\,\delta\mathbf{x}\| \leq c\|\delta\mathbf{B}\| \tag{8}$$

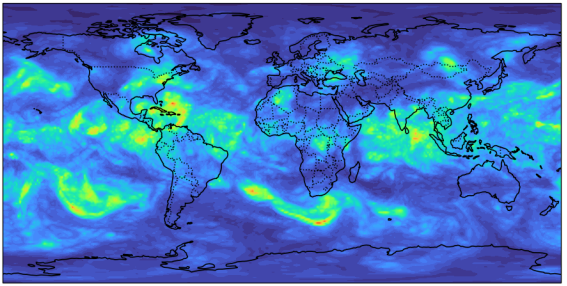
where $d\,\delta\mathbf{x}$ denotes *change in analysis increment*.

# Experiment results – vo mlev74 20220601 1800

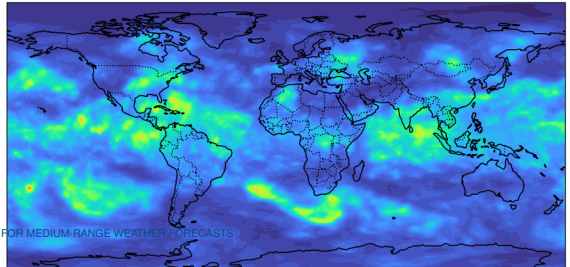
5-EDA-es



es

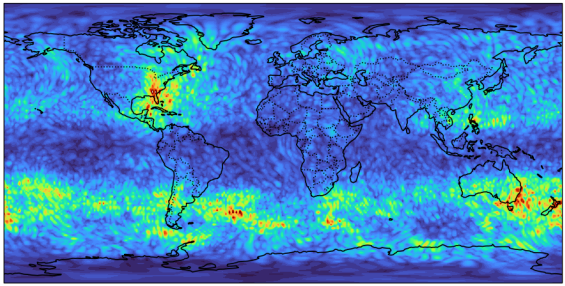


ML-es

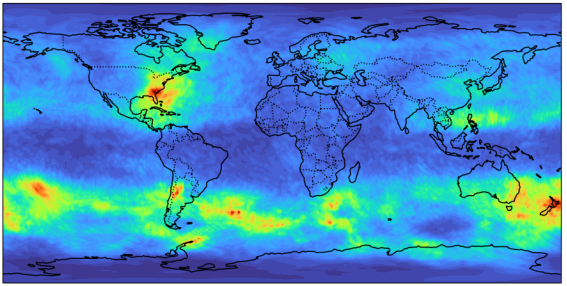


# Experiment results – ucdv mlev02 20220601 1800

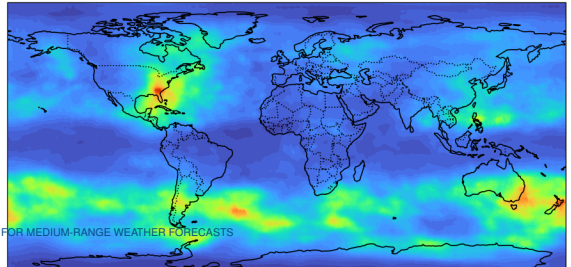
5-EDA-es



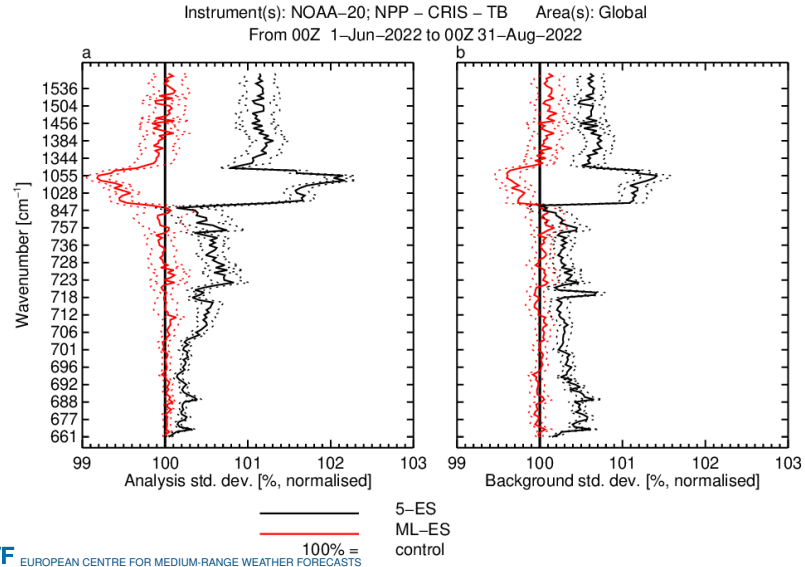
es



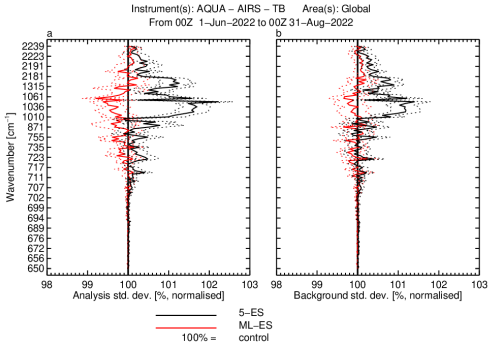
ML-es

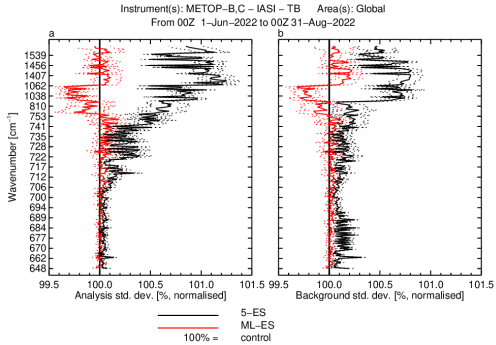


# Experiment results – (iver) ref exp 50r1 control

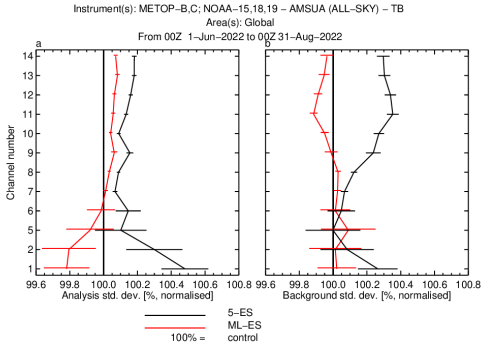


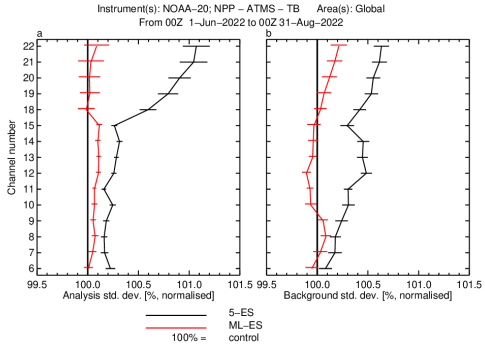
# Experiment results – (iver) ref exp 50r1 control



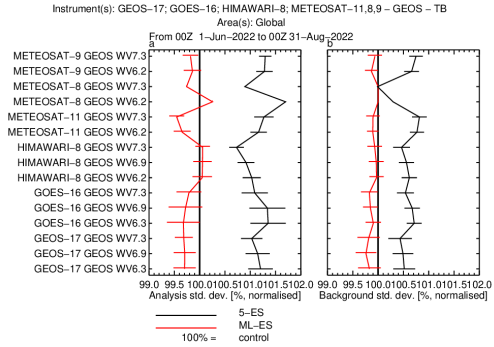



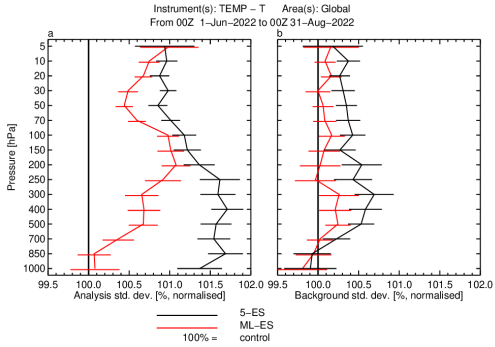
# Experiment results – (iver) ref exp 50r1 control



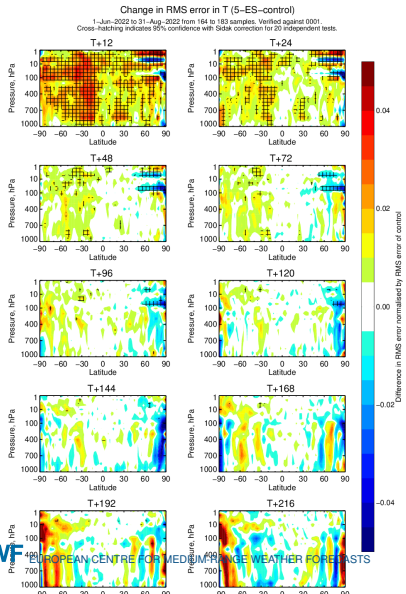



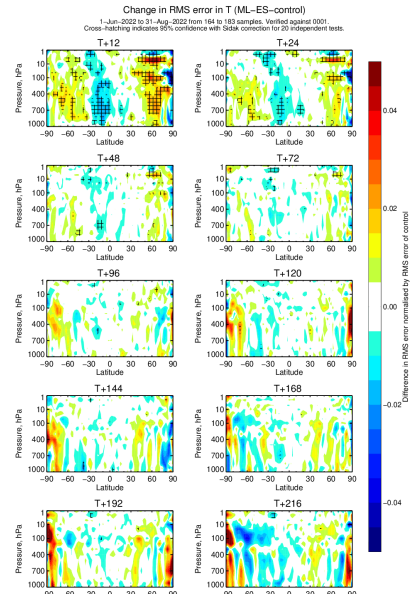
# Experiment results – (iver) ref exp 50r1 control



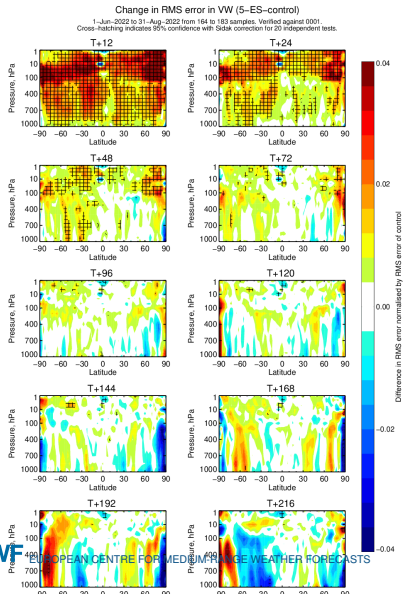



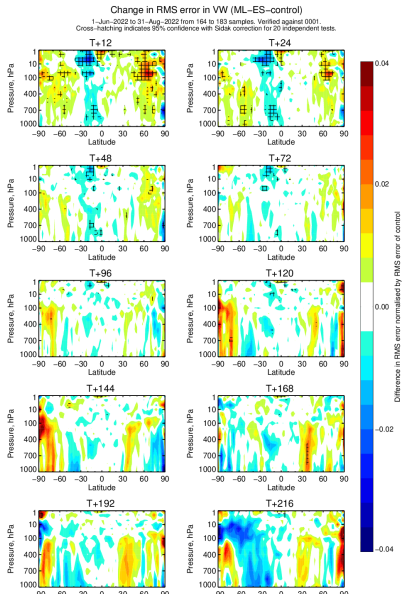
# Experiment results – (iver) ref exp 50r1 control - T fc



# Experiment results – (iver) ref exp 50r1 control - wind fc



# Perturbations generator

The perturbations generator is a parametric conditional probability distribution

$$\nu_\theta(\mathbf{x}', t \,|\, \mathbf{x}_1'^a, \ldots, \mathbf{x}_M'^a) \quad \xrightarrow{\text{sample}} \quad \mathbf{x}'^a. \tag{9}$$

which we view as the solution to a specific instance of the following problem.

## Problem

*Let $\mathfrak{X}(\mathcal{M})$ denote the space of vector fields on a smooth manifold $\mathcal{M}$. Let $\mathcal{P}(\mathfrak{X}(\mathcal{M}))$ denote the space of probability measures on $\mathfrak{X}(\mathcal{M})$. Let $D$ be a metric on $\mathcal{P}(\mathfrak{X}(\mathcal{M}))$.*

*Given $\mu, \tilde{\mu} \in \mathcal{P}(\mathfrak{X}(\mathcal{M}))$ with $\tilde{\mu} \ll \mu$, find $\nu* \in \mathcal{P}(\mathfrak{X}(\mathcal{M}))$ such that*

$$\nu* = \underset{\substack{\nu \in \mathcal{P}(\mathfrak{X}(\mathcal{M})) \\ \nu \ll \mu}}{\arg\min} \; D(\,\mu,\, \alpha\tilde{\mu} + (1-\alpha)\nu\,), \tag{10}$$

*for some $\alpha \in (0, 1)$.*

# Perturbations generator

ETKF for Lorenz96 standard setup: $F = 8$, $N_x = 40$

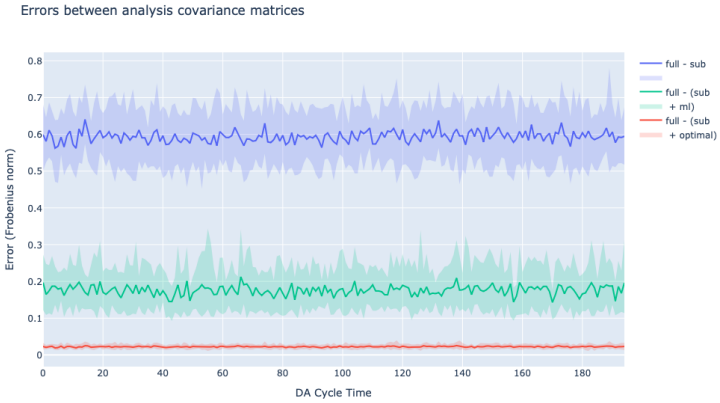
ETKF analysis perturbs ensemble:

$$x_i'^a(t), \quad i = 1, \ldots, 20$$

Subset-ensemble
+ ML generated perturbs:

$$x_i'^a(t), \quad i = 1, \ldots, 10$$
and $\tilde{x}_j'^a(t), \quad j = 11, \ldots 20$

Compare analysis covariance matrices $P^a_{\text{full}}(t)$, $P^a_{\text{sub}}(t)$ and $P^a_{\text{sub+ML}}(t)$



# Summary and outlook

- First results (statistics emulator) sets a benchmark for further improvements.
- More tests and evaluations are in the pipeline.
- Ensemble correlations.
- Continued development of the perturbations generator.