Latent-space data assimilation and background-error covariance modelling

ECMWF Annual Seminar 2026

Žiga Zaplotnik[1] **,** Boštjan Melinc[2], Uroš Perkan[2] and Lukas Kugler[2]
[1]ECMWF
[2]University of Ljubljana, Faculty of Mathematics and Physics

ECMWF

© ECMWF September 17, 2026

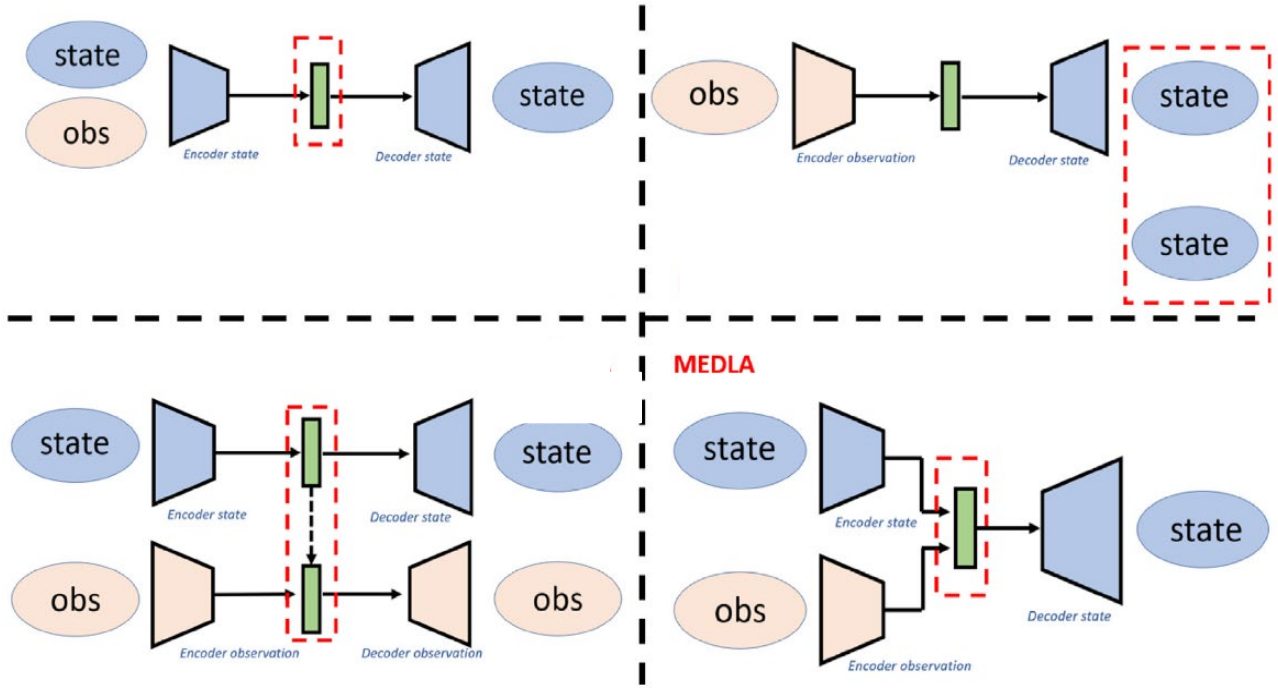
Content

1) An overview of different LSDA approaches
2) Motivation for LSDA
3) 3D-Var in the latent space
4) Results/examples:
   - Proof of work: Bostjan's second PhD paper on 3D-Var with multilevel multivariate convolutional autoencoders
   - GNN for LSDA
5) Conclusions
6) "Did you enjoy the weather last summer?"

An Overview of Latent Space Data Assimilation approaches

- Incremental 3D-Var (Mack et al., 2020)
- Ensemble 3D-Var (Melinc and Zaplotnik, 2024; Melinc et al., 2026)
- 4D-Var (Fan et al., 2026a)
- Kalman filter (Amendola et al., 2021; Buchnik et al. 2023; Falconer et al., 2025)
- Ensemble Kalman filter (Peyron et al., 2021; Pasmans et al., 2025; Fan et al., 2026b)
- Non-conventional schemes (Cheng et al., 2023; Cheng et al., 2024)
- Different options to handle observations and background (Fan et al., 2025a,b):
  - Keep using observation operators or encode observations to shared latent space?
  - Omit using background?

Different conceptual flavours of LSDA

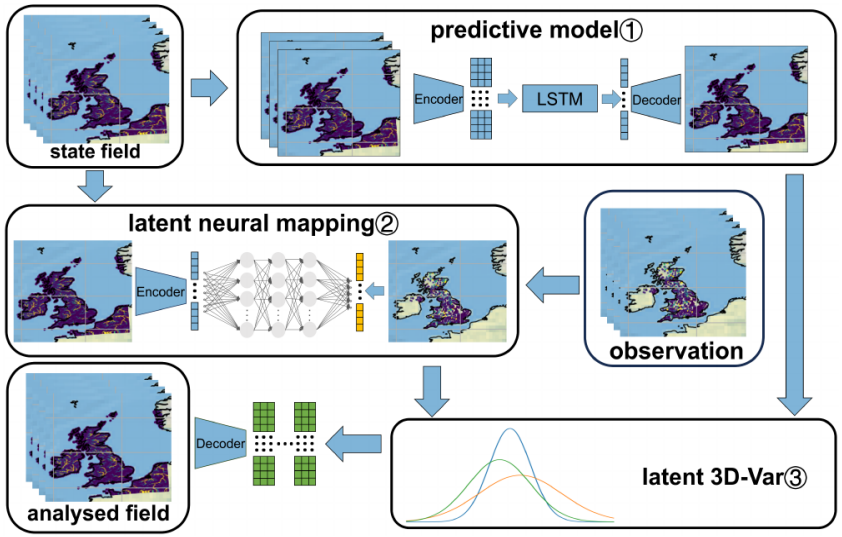


Cheng et al. (2024)

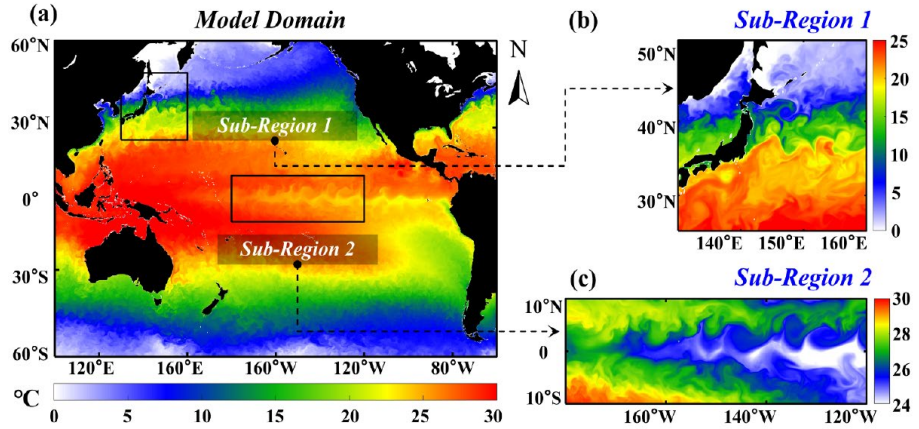
indicates where DA step takes place

Latent space DA applications so far (beyond toy models)

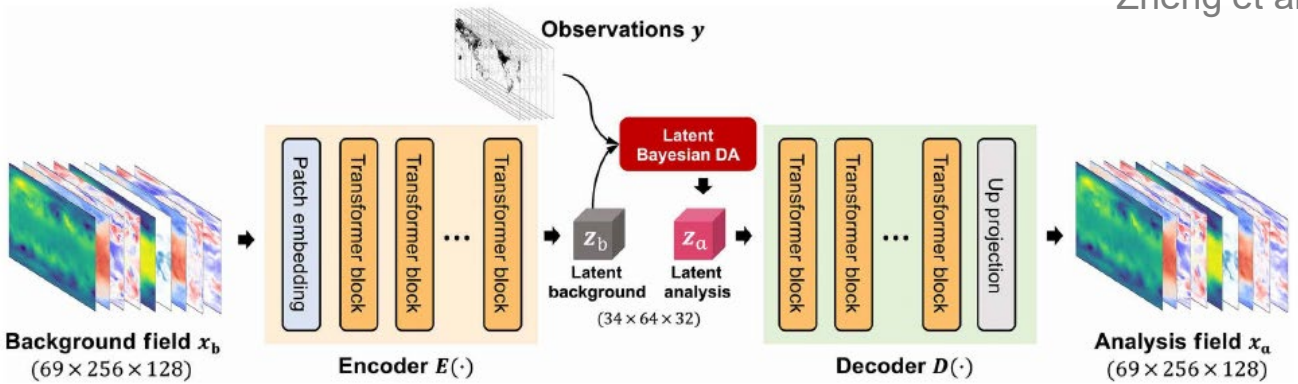
- Air-quality models (Mack et al., 2020; Amendola et al., 2021)
- Global atmospheric models (Melinc and Zaplotnik, 2024; Melinc et al., 2026; Fan et al., 2026a,b; Zaplotnik et al., 2026)
- Regional atmospheric models (Fan et al., 2025a,b)
- River-discharge (Wang et al., 2025)
- Regional SST model (Zheng et al., 2026)



Wang et al. (2025)



Zheng et al. (2026)



Fan et al. (2026a)

Motivation

1) Reduce cost and speed-up minimisation:

Operational DA requires trade-offs between mathematical rigor and computational efficiency

Example: 3D-Var cost function

$$\mathcal{J}(\mathbf{x}) = \mathcal{J}_b + \mathcal{J}_o = \\ = \frac{1}{2}(\mathbf{x}-\mathbf{x}_b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_b) + \frac{1}{2}\{\mathbf{y}-H(\mathbf{x})\}^{\mathrm{T}}\mathbf{R}^{-1}\{\mathbf{y}-H(\mathbf{x})\}$$

Challenge: **x** ~ 10^9 elements → unconstrained **B** ~ $10^9 \times 10^9$ elements

Solution:

- Variational DA (3D/4D-Var) cost function minimisation is performed in a decorrelated **control** space, typically reduced with respect to model space
- can we produce DA in an even smaller space?

Motivation

2) Improve the representation of background-error covariances

Currently, at ECMWF, we transform the minimisation of the classic 3D-var cost function:

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{x}_b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_b) + \frac{1}{2}\{\mathbf{y}-H(\mathbf{x})\}^{\mathrm{T}}\mathbf{R}^{-1}\{\mathbf{y}-H(\mathbf{x})\}$$

into minimisation of

$$\mathcal{J}_\chi(\chi) = \frac{1}{2}\chi^{\mathrm{T}}\chi + \frac{1}{2}\{\mathbf{y}-H(\mathbf{x}_b+\mathbf{L}\chi)\}^{\mathrm{T}}\mathbf{R}^{-1}\{\mathbf{y}-H(\mathbf{x}_b+\mathbf{L}\chi)\}$$

by applying the following control variable transform

$$\delta\mathbf{x} = (\mathbf{x}-\mathbf{x}_b) = \mathbf{K}\boldsymbol{\Sigma}_b^{1/2}\sum_j \psi_j \otimes \left[\mathbf{C}_j^{1/2}(\lambda,\phi)\chi_j\right]$$

$\delta\mathbf{x} = \mathbf{L}\boldsymbol{\chi}$ such that $\mathbf{B} = \mathbf{L}\mathbf{L}^T$

Motivation

2) Improve the representation of background-error covariances

The balance transform $\mathbf{K}$ maps unbalanced control variable $[\delta\tilde{\zeta},\ \delta\tilde{\eta}_u,\ (\delta\tilde{T},\delta\tilde{p}_s)_u,\ \delta\tilde{q},\ \delta\tilde{r}(O_3),\ldots]^T$ to model variables $[\delta\zeta,\ \delta\eta,\ (\delta T,\delta p_s),\ \delta q,\ \delta r(O_3),\ldots]^T$ by doing

$$
\begin{aligned}
\delta\zeta &= \delta\tilde{\zeta} \\
(\delta T,\delta p_s) &= \underbrace{\mathcal{N}\mathcal{H}\,\delta\tilde{\zeta}}_{(\delta T,\delta p_s)_{bal}(\delta\Phi_{bal}(\delta\zeta))} + \mathcal{P}\,\delta\tilde{\eta}_u + (\delta\tilde{T},\delta\tilde{p}_s)_u \\
\delta\eta &= \underbrace{\mathcal{M}\mathcal{H}\,\delta\tilde{\zeta}}_{\delta\eta_{bal}(\delta\Phi_{bal}(\delta\zeta))} + \mathcal{Q}_\zeta\,\delta\tilde{\zeta} + \mathcal{Q}_T\,\delta T + \delta\tilde{\eta}_u \\
\delta q &= q_s(T^b)\,\delta\tilde{q}_u + Q_q T\frac{q^b L}{R_v(T^b)^2}\delta T \\
\delta r(O_3) &= \mathcal{S}(\delta\tilde{\zeta}) + \delta\tilde{r}_u(O_3)
\end{aligned}
$$

$\mathcal{M}$, $\mathcal{N}$ are vertical balance operators

$\mathcal{H}$ is a horizontal balance operator based on nonlinear balance equation

$\mathcal{Q}_\zeta$, $\mathcal{Q}_T$ are balanced divergence operators based on QG ω-equation

Motivation

2) Improve the representation of background-error covariances

The balance transform $\mathbf{K}$ maps unbalanced control variable $[\delta\tilde{\zeta},\ \delta\tilde{\eta}_u,\ (\delta\tilde{T},\delta\tilde{p}_s)_u,\ \delta\tilde{q},\ \delta\tilde{r}(O_3),\ldots]^T$ to model variables $[\delta\zeta,\ \delta\eta,\ (\delta T,\delta p_s),\ \delta q,\ \delta r(O_3),\ldots]^T$ by doing

$$\delta\zeta = \delta\tilde{\zeta}$$

These balance operators include series of linear regressions (in the spectral space).

→ The balance transform (or a CVT as a whole) is a natural candidate to be replaced by just another regression, the neural network.

$\mathcal{M}$, $\mathcal{N}$ are vertical balance operators

$\mathcal{H}$ is a horizontal balance operator based on nonlinear balance equation

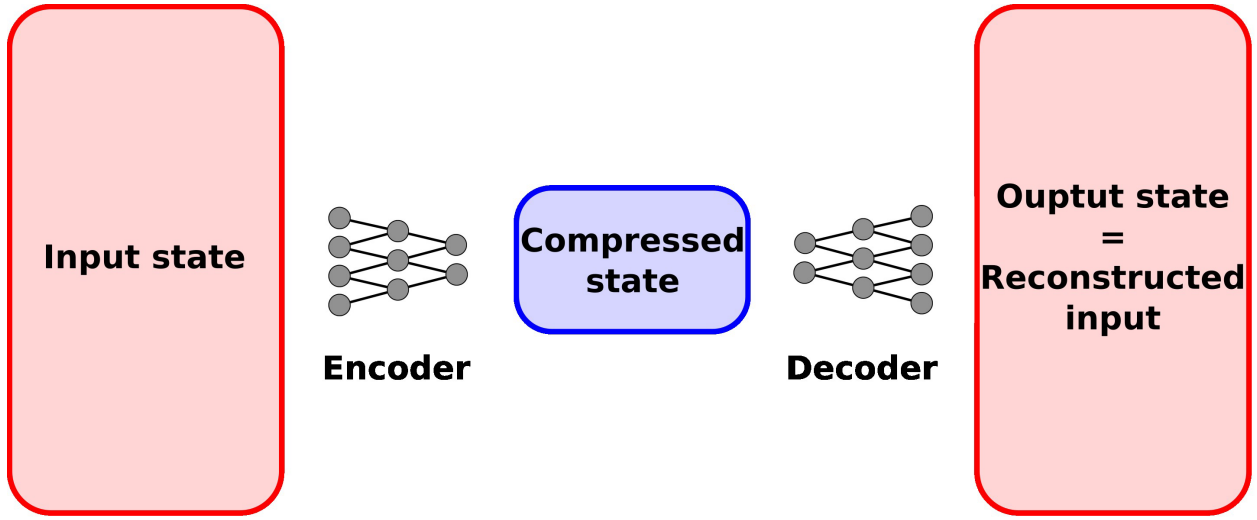
$\mathcal{Q}_\zeta$, $\mathcal{Q}_T$ are balanced divergence operators based on QG ω-equation

Representation of the atmospheric (and/or ocean, land-surface) state using an Autoencoder

Autoencoder (AE) is a neural network (NN), trained to reconstruct its input with intermediate compression

Semantics:

- compressed space = ***latent space***
- compressed state = ***latent state***



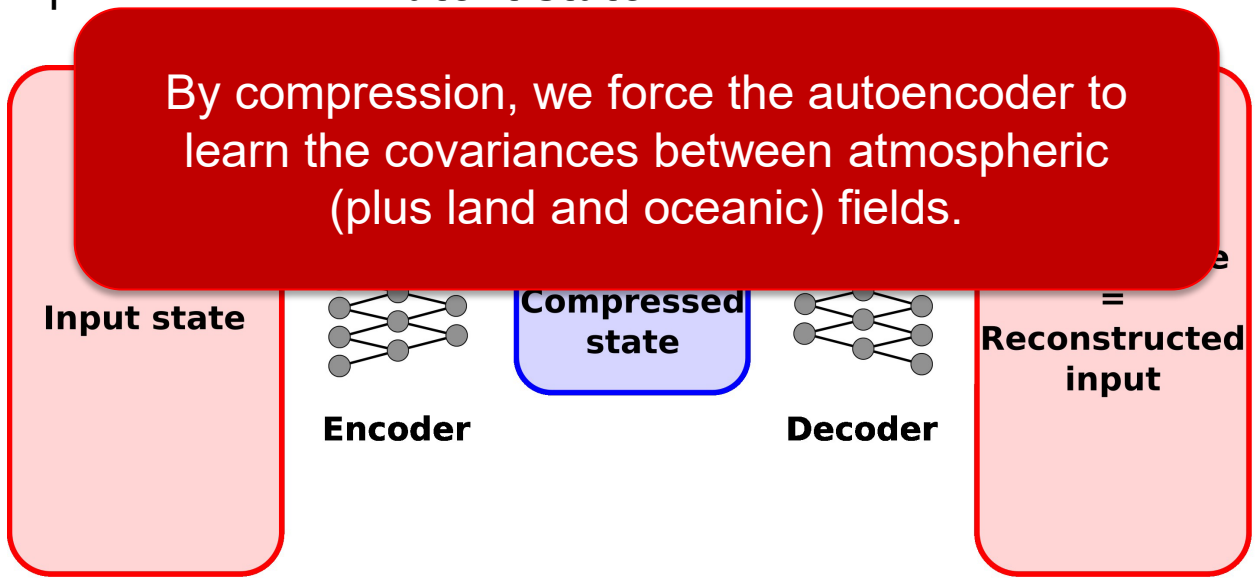
$$x \xrightarrow{E} z=E(x) \xrightarrow{D} x'=D(z) = D(E(x))$$

Representation of the atmospheric (and/or ocean, land-surface) state using an Autoencoder

Autoencoder (AE) is a neural network (NN), trained to reconstruct its input with intermediate compression

Semantics:

- compressed space = ***latent space***
- compressed state = ***latent state***



$$\mathbf{x} \xrightarrow{E} \mathbf{z}=E(\mathbf{x}) \xrightarrow{D} \mathbf{x}'=D(\mathbf{z}) = D(E(\mathbf{x}))$$

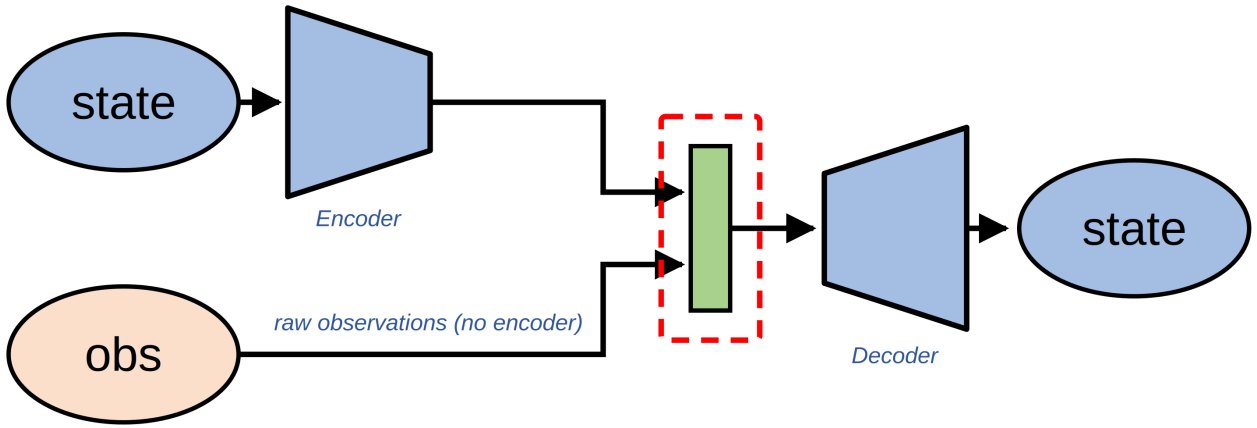
3D-Var in a latent space of an AE (LS3D-Var)

Assumptions (similar as conventional 3D-Var):

- Background and observation errors are independent
- Their errors are Gaussian

Cost function:

$$\mathcal{J}_z(\mathbf{z}) = \mathcal{J}_{bz} + \mathcal{J}_{oz} =$$
$$= \frac{1}{2}(\mathbf{z} - \mathbf{z}_b)^{\mathrm{T}}\mathbf{B}_z^{-1}(\mathbf{z} - \mathbf{z}_b) + \frac{1}{2}[\mathbf{y} - H\{D(\mathbf{z})\}]^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{y} - H\{D(\mathbf{z})\}]$$



z … latent vector
$\mathbf{z}_b$ … background defined in latent space
$\mathbf{B}_z$ … background-error covariance matrix
y … observation vector
H … observation operator
D … decode and destandardise
R … observation-error covariance matrix

Practical example: LS3D-Var using a global mutlivariate multilevel atmospheric DA model using a CNN AE and a UNet forecasting model

CNN Autoencoder

- Goal: reconstruct 20 global input fields at 1° resolution on regular grid
- Training data: ERA5 reanalyses
  (training set: 1970-2014, validation set: 2015-2019, test set: 2020-2023)

Quantity [unit]	Levels	Abbreviation
Geopotential height [m]	250 hPa, 500 hPa, 700 hPa, 850 hPa	Z250, Z500, Z700, Z850
Zonal wind [m/s]	200 hPa, 500 hPa, 700 hPa, 900 hPa, 10 m	U200, U500, U700, U900, U10m
Meridional wind [m/s]	200 hPa, 500 hPa, 700 hPa, 900 hPa, 10 m	V200, V500, V700, V900, V10m
Temperature [K]	500 hPa, 850 hPa, 2 m, surface	T500, T850, T2m, ST
Mean sea level pressure [hPa]	-	MSLP
Total column water vapor [kg/m^2]	-	TCWV

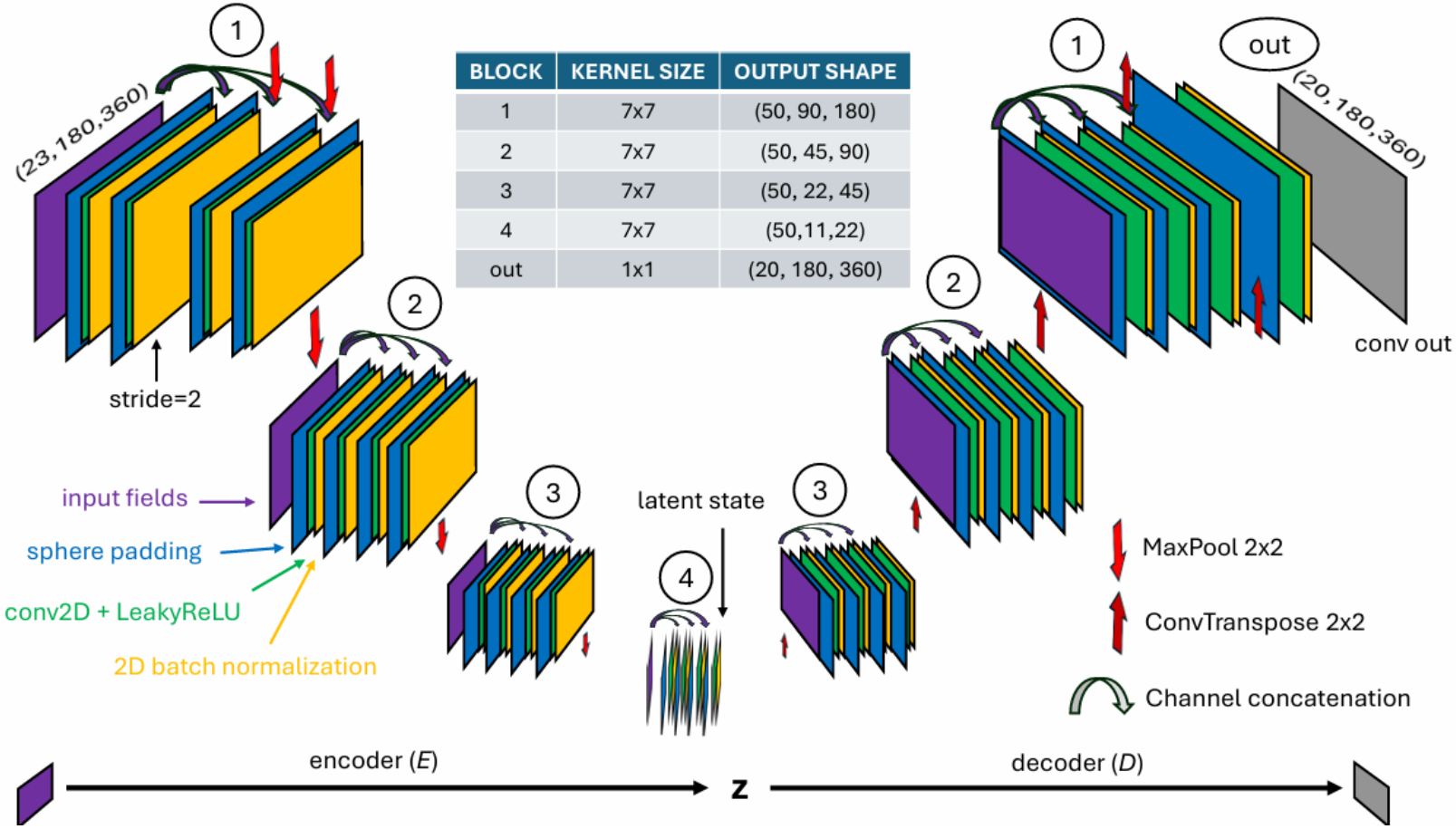
Section based on
Melinc, Perkan and Zaplotnik (2026), JAMES

Practical example: LS3D-Var using a global mutlivariate multilevel atmospheric DA model using a CNN AE and a UNet forecasting model

AE structure:

- **Input tensor**:
  23×180×360 = **1.5M features**
  (20 atm. fields + 3 static fields)
- Convolutional blocks
- **Latent vector**:
  12k features
  (2 orders of magnitude reduction!)
- Output tensor:
  20×180×360 = 1.3M features

BLOCK	KERNEL SIZE	OUTPUT SHAPE
1	7x7	(50, 90, 180)
2	7x7	(50, 45, 90)
3	7x7	(50, 22, 45)
4	7x7	(50,11,22)
out	1x1	(20, 180, 360)



Practical example: LS3D-Var using a global mutlivariate multilevel atmospheric DA model using a CNN AE and a UNet forecasting model

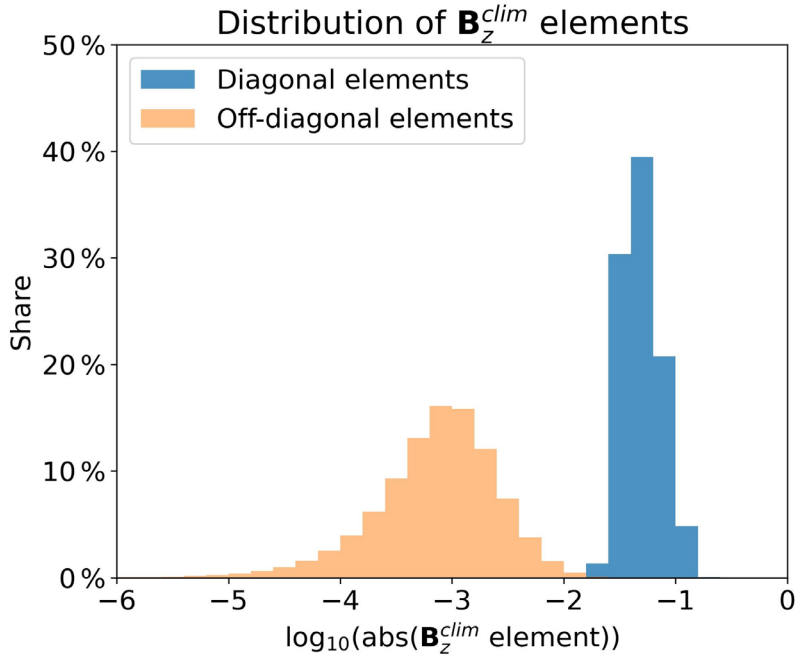
$$\mathcal{J}_z(\mathbf{z}) = \mathcal{J}_{bz} + \mathcal{J}_{oz} =$$
$$= \frac{1}{2}(\mathbf{z}-\mathbf{z}_b)^{\mathrm{T}}\mathbf{B}_z^{-1}(\mathbf{z}-\mathbf{z}_b) + \frac{1}{2}[\mathbf{y}-H\{D(\mathbf{z})\}]^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{y}-H\{D(\mathbf{z})\}]$$

Background-error covariance modelling:

- Climatological $\mathbf{B}_z$-matrix:

$$\mathbf{B}_z^{clim} = \left\langle \left(\mathbf{z}_b(t)-\mathbf{z}_t(t)\right)\left(\mathbf{z}_b(t)-\mathbf{z}_t(t)\right)^{\top} \right\rangle$$
$$\mathbf{z}_b(t) = E\left(\mathrm{NN}^{24\mathrm{h}}[\mathbf{x}_{\mathrm{ERA5}}(t-24\mathrm{h})]\right)$$
$$\mathbf{z}_t(t) = E\left(\mathbf{x}_{\mathrm{ERA5}}(t)\right)$$

- Average performed over validation set (2015-2019)
- Quasi-diagonality assumption based on scale-separation of diagonal and off-diagonal elements
  → Use only diagonals for inverse computation



Practical example: LS3D-Var using a global mutlivariate multilevel atmospheric DA model using a CNN AE and a UNet forecasting model

Convergence and gradient evaluation:

- To achieve faster cost function convergence, we precondition it:

$$\begin{aligned}\mathcal{J}_\chi(\boldsymbol{\chi}) &= \frac{1}{2}\boldsymbol{\chi}^\top\boldsymbol{\chi} \\ &+ \frac{1}{2}\left[\mathbf{y} - H\left\{D\left(\mathbf{z}_b + \mathbf{L}_z\boldsymbol{\chi}\right)\right\}\right]^\top \mathbf{R}^{-1}\left[\mathbf{y} - H\left\{D\left(\mathbf{z}_b + \mathbf{L}_z\boldsymbol{\chi}\right)\right\}\right],\end{aligned}$$

  where $\boldsymbol{\chi} = \mathbf{L}_z^{-1}(\mathbf{z} - \mathbf{z}_b)$ and $\mathbf{L}_z = \mathbf{B}_z^{1/2}$.

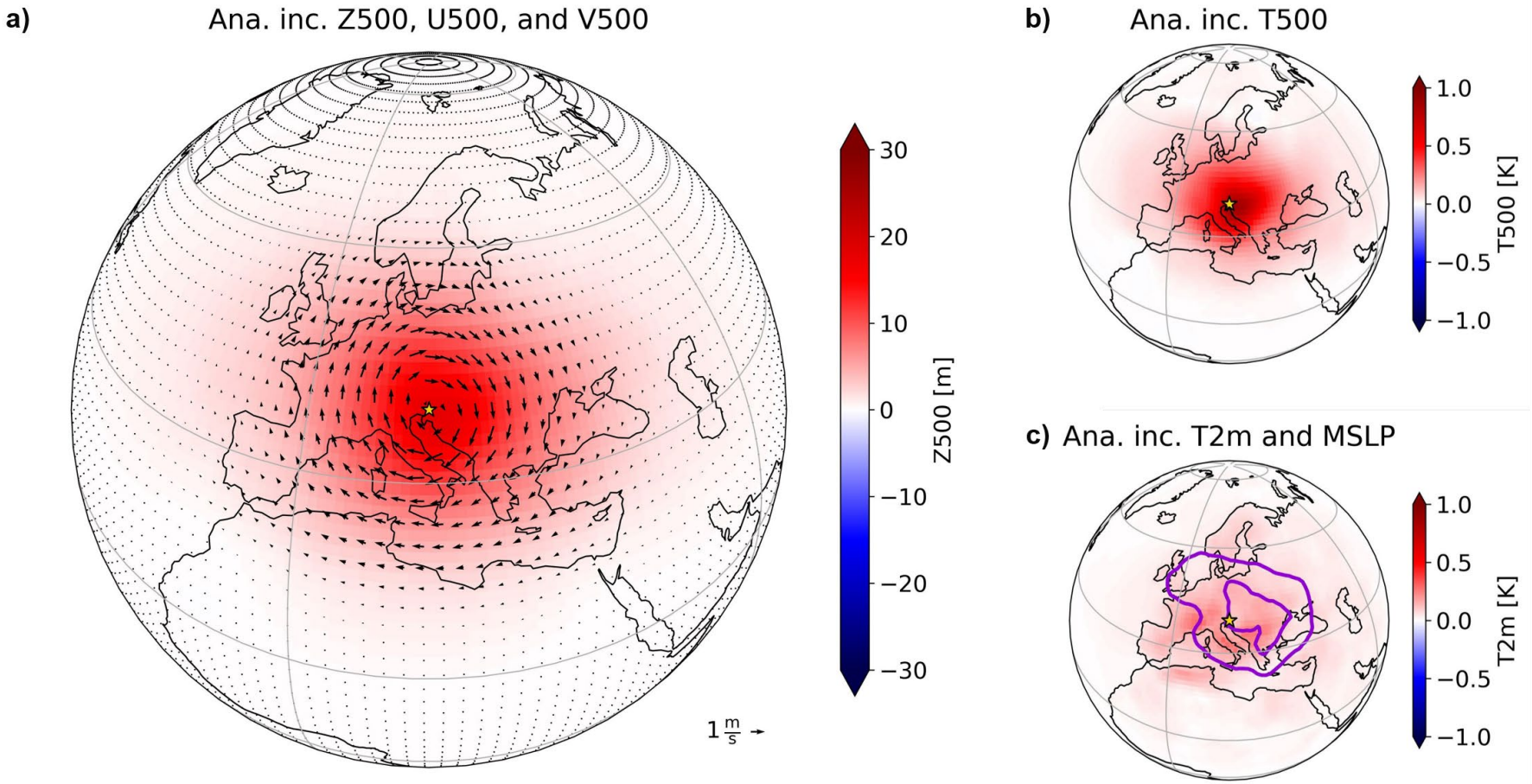
- Convergence criterion based on $\|\nabla\mathcal{J}_\chi\|$
- As long as every element of the cost function is written using Pytorch's autodifferentiable functions, gradient evaluation becomes very cheap!

Practical example – midlatitude balances

Observing Z500 above Ljubljana, Slovenia (46.1˚N, 14.5˚E): d^{Z500} = +30 m, σ_o = 10 m

Results in:

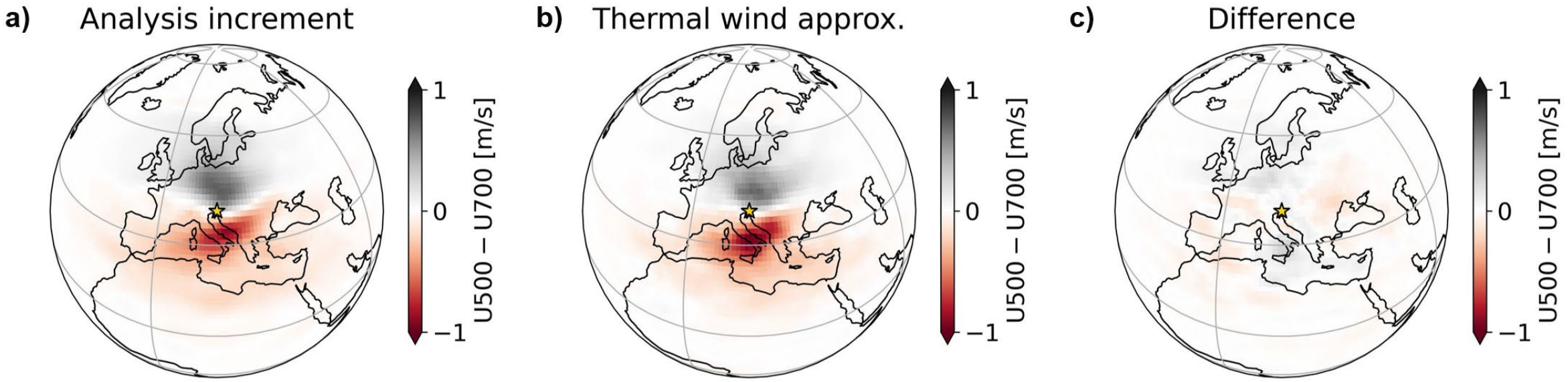
- Positive Z500 increment
- Large-scale balances:
  - Anticyclonic wind inc. (geostrophic balance)
  - T500 inc. coincides with Z500 increment
  - MSLP inc. peaks eastward of obs. loc.
- Local features:
  - T2m inc. confined to land
  - MSLP inc. distorted due to orography



Practical example – thermal wind balance

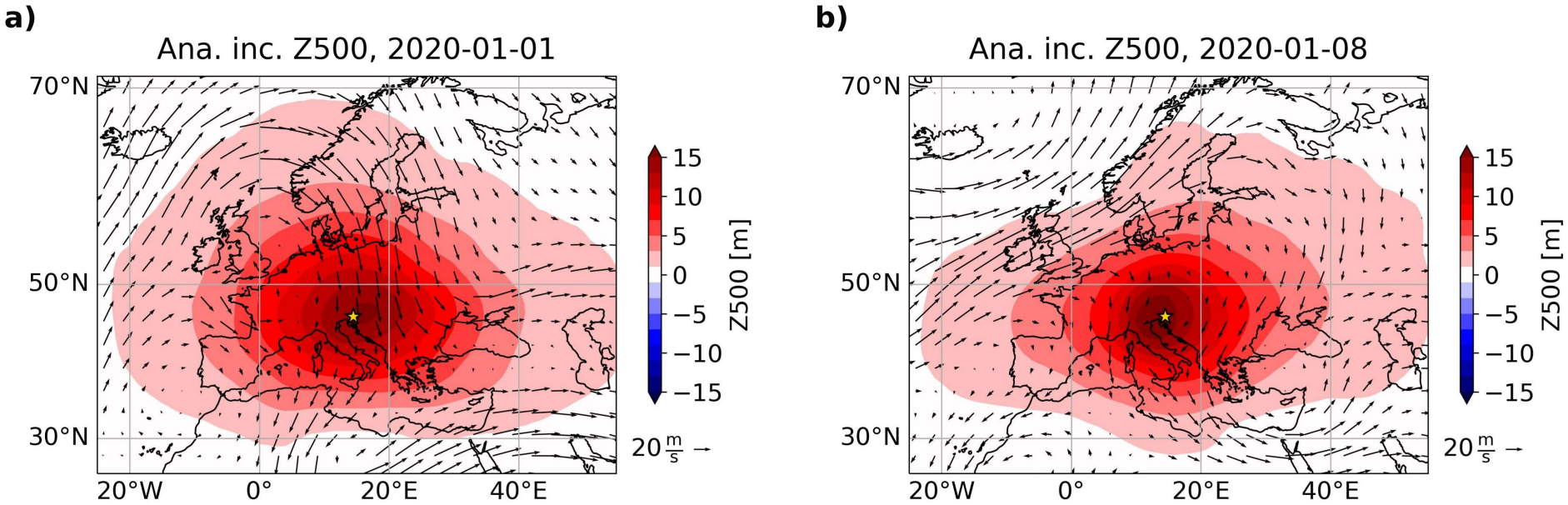
If both **geostrophic** and **hydrostatic balances** are obeyed, **thermal wind** approximation should hold:

$$\delta_a^{\mathrm{U500}} - \delta_a^{\mathrm{U700}} \approx -\frac{g}{f}\frac{\partial(\delta_a^{\mathrm{Z500}} - \delta_a^{\mathrm{Z700}})}{\partial y}$$



Practical example – flow dependence

Despite using static climatology-based **B**-matrix (in the latent space), the analysis increments (in physical space) show **flow-dependent** behavior



Reason: decoder's nonlinearity. Let $\delta\mathbf{z}$ be a *small* perturbation, that we add to two *distant* latent vectors $\mathbf{z}_1$ and $\mathbf{z}_2$. (i.e., apart enough that D cannot be treated as affine in the interim area). The respective changes of their decoded counterparts are

$$\delta\mathbf{x}_1 = D(\mathbf{z}_1 + \delta\mathbf{z}) - D(\mathbf{z}_1) \approx D(\mathbf{z}_1) + \left.\frac{\partial D}{\partial \mathbf{z}}\right|_{\mathbf{z}_1} \delta\mathbf{z} - D(\mathbf{z}_1)$$

$$\delta\mathbf{x}_1 \approx \left.\frac{\partial D}{\partial \mathbf{z}}\right|_{\mathbf{z}_1} \delta\mathbf{z}$$

and

$$\delta\mathbf{x}_2 \approx \left.\frac{\partial D}{\partial \mathbf{z}}\right|_{\mathbf{z}_2} \delta\mathbf{z}$$

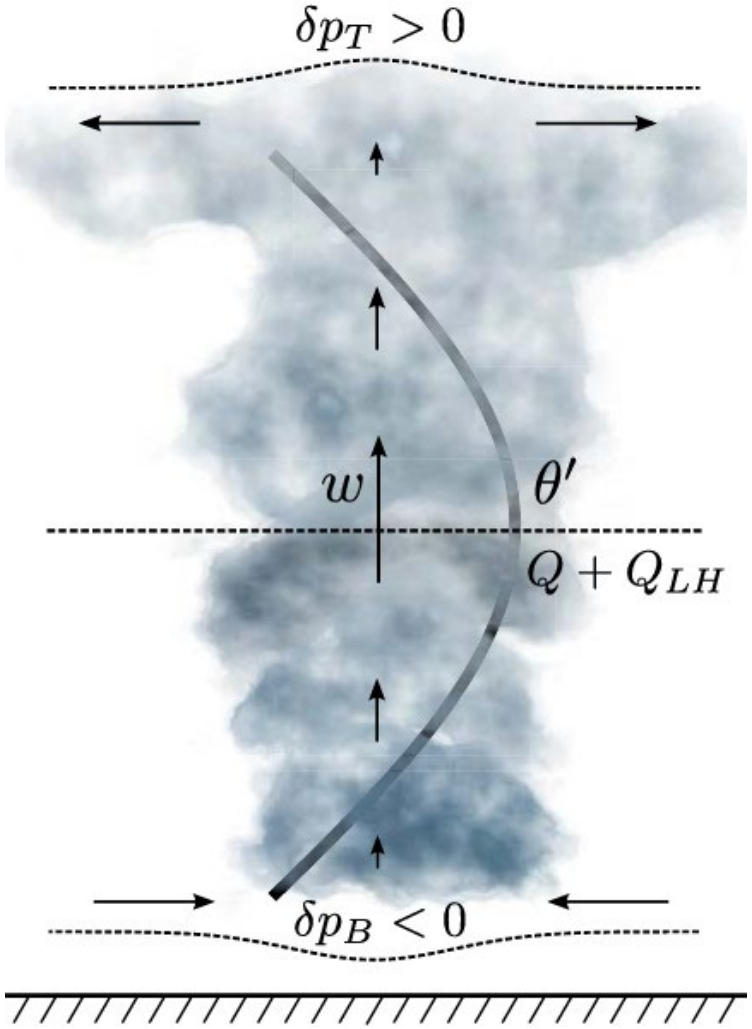
Practical example – tropical balances

Observing TCWV in nearly saturated area in Central Atlantic (0˚N, 33˚W)

- d^{TCWV} = +10 kg/m^2
- σ_o = 3 kg/m^2

Theory: heat-induced tropical perturbations
(Gill, 1980; Davey and Gill, 1987):

condensation → latent heat release → updraft →
→ lower tropospheric convergence & upper tropospheric divergence

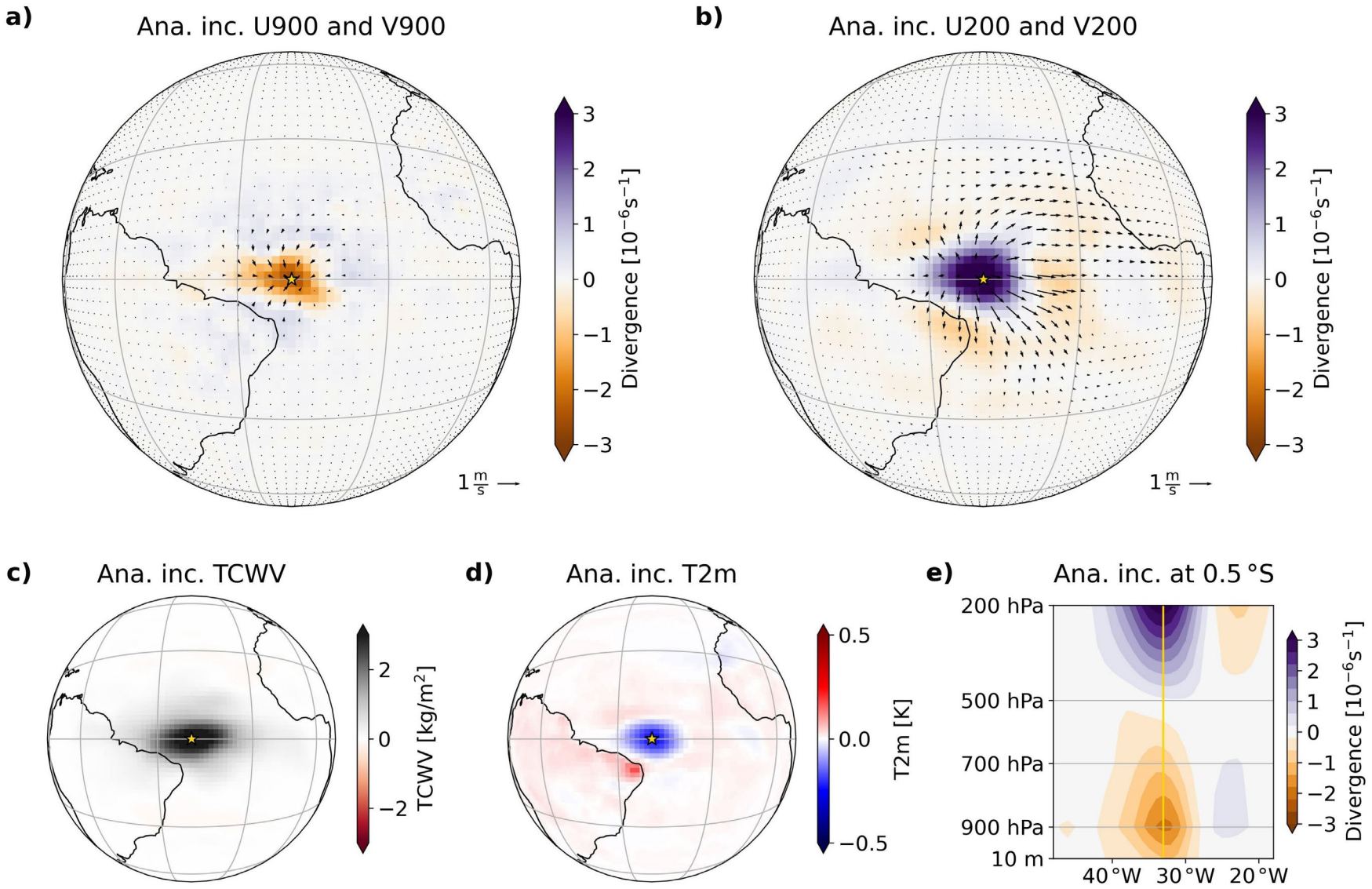


Practical example – tropical balances

Analysis increments obey the theory long-envisaged by A. Gill (1980)

Note: the processes (condensation, precipitation, latent heat release, and vertical wind response) are not explicitly fed into autoencoder!

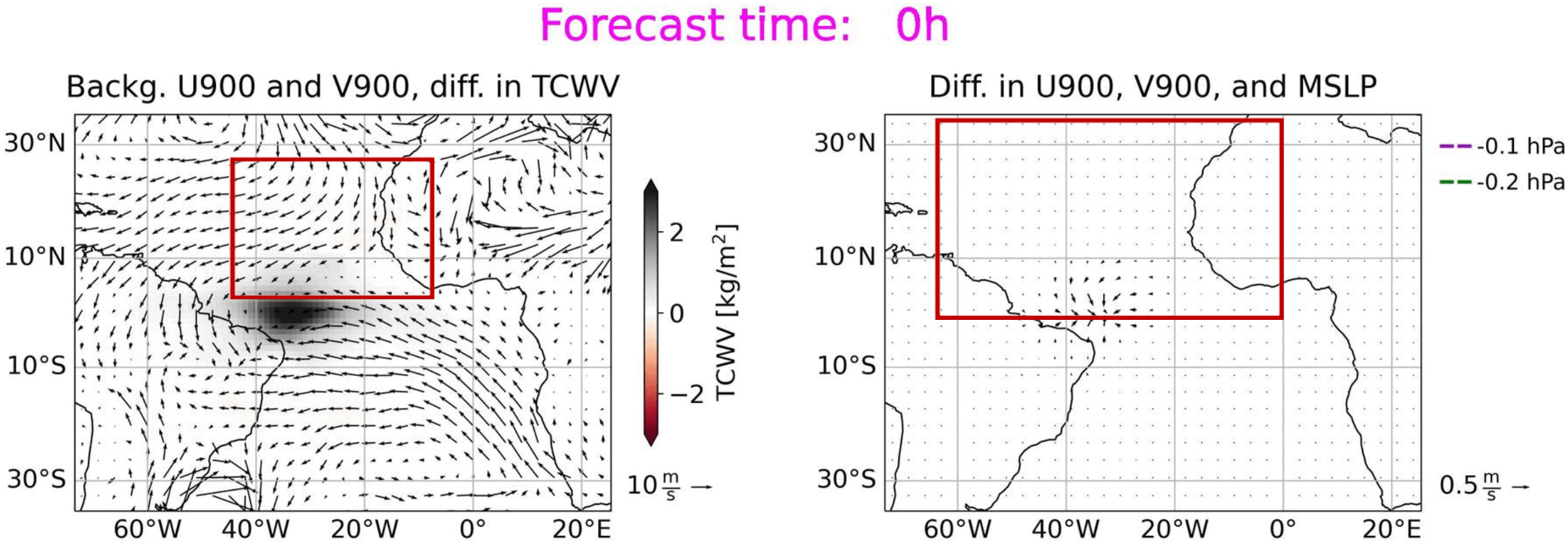
→ these balances are learnt implicitly through other variables



Practical example – tropical balances

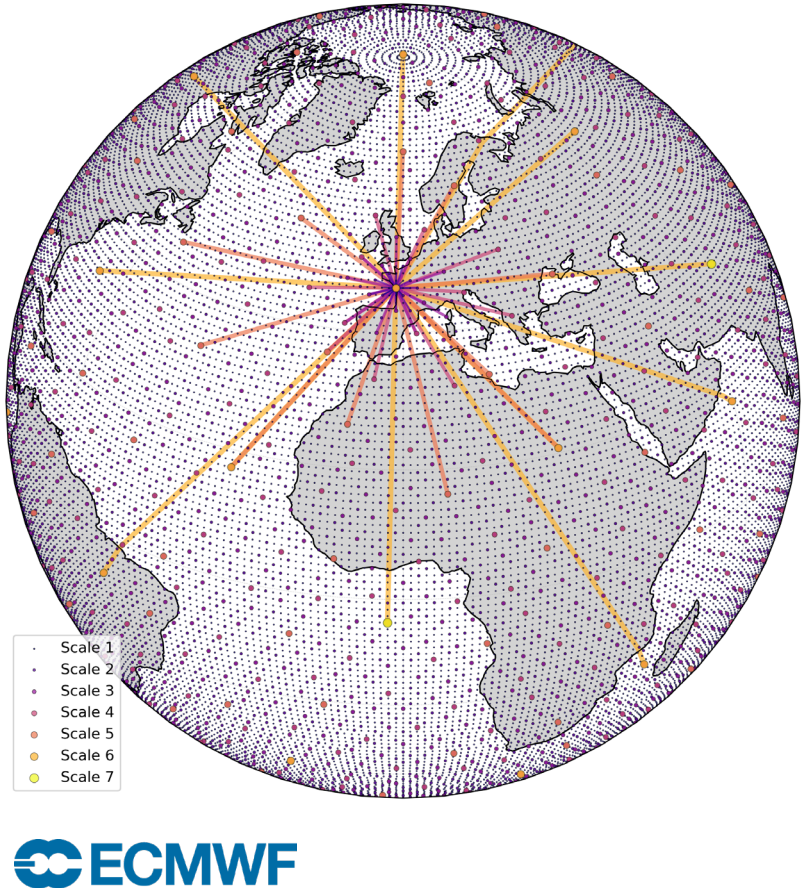
Difference between respective forecasts initialised by the analysis and the background:

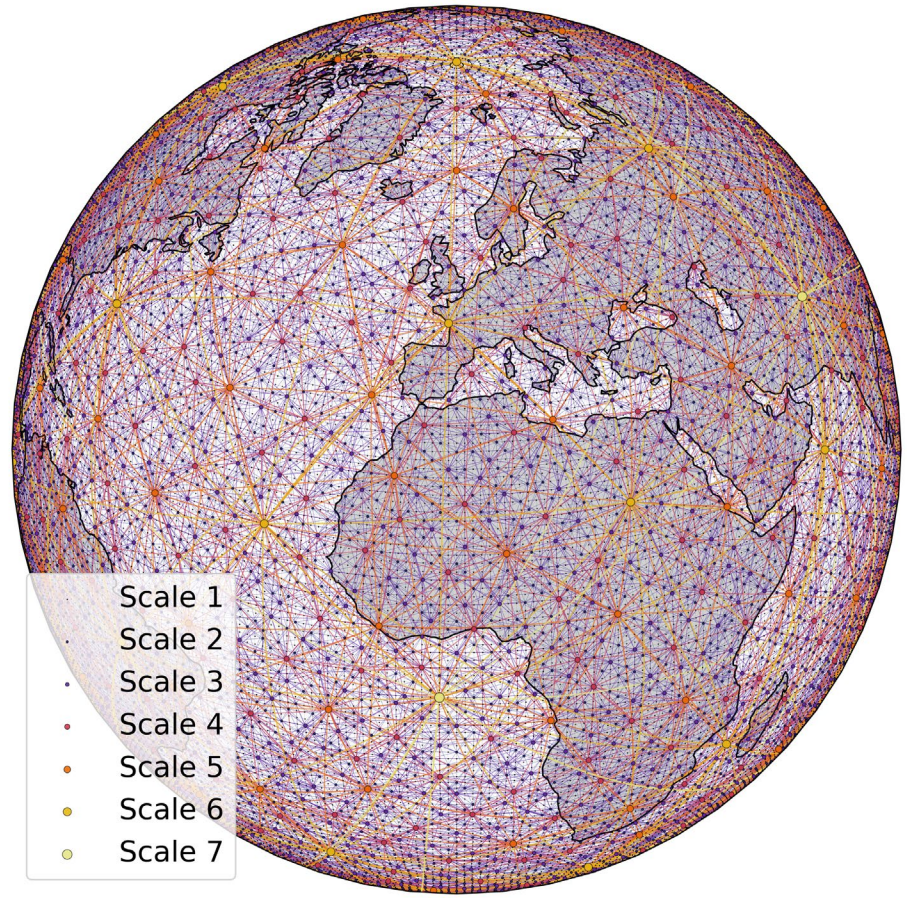
- TCWV increment advected by lower-tropospheric wind
- Eastward propagating Kelvin wave



GNN transformer β –VAE on a multiscale graph

- Experimental setup: 18 pressure levels, N80 resolution
- 6 atmospheric variables (T, u ,v, Z, q, r(O_3)) at each level
- 8 surface variables (sst, t2m, mslp, stl1, siconc, sd) + 6 static features

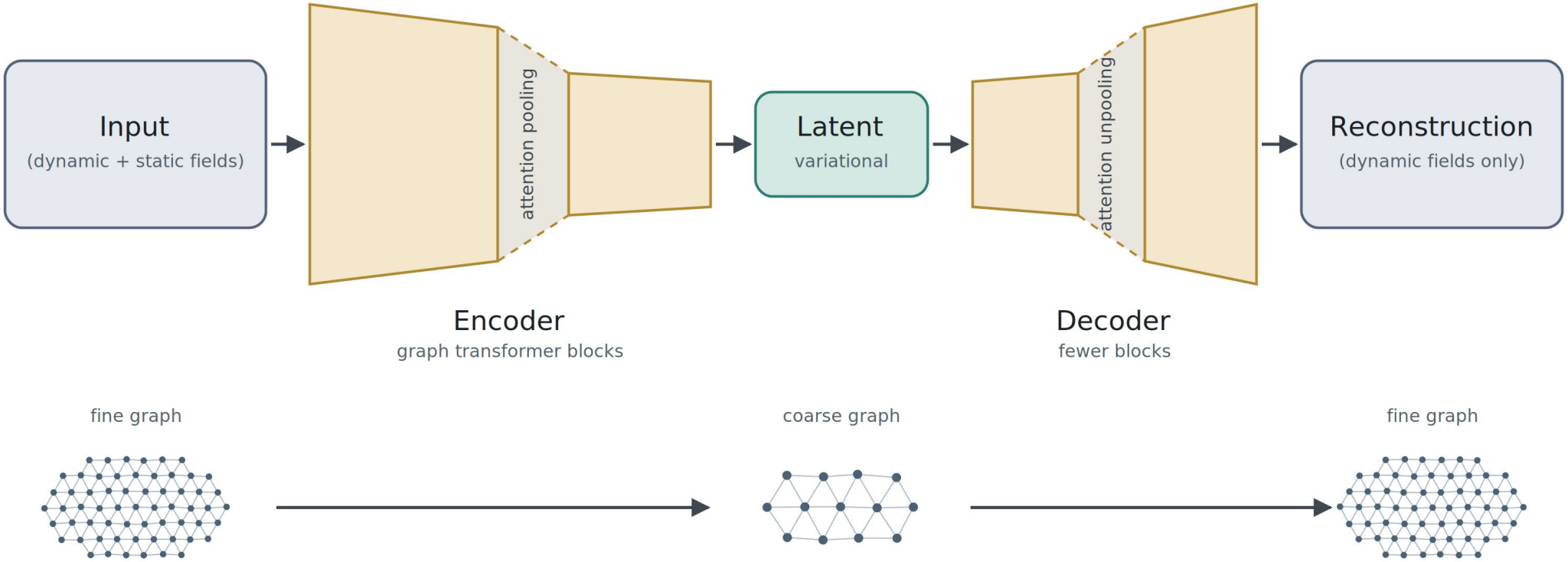




ECMWF

GNN transformer β –VAE on a multiscale graph

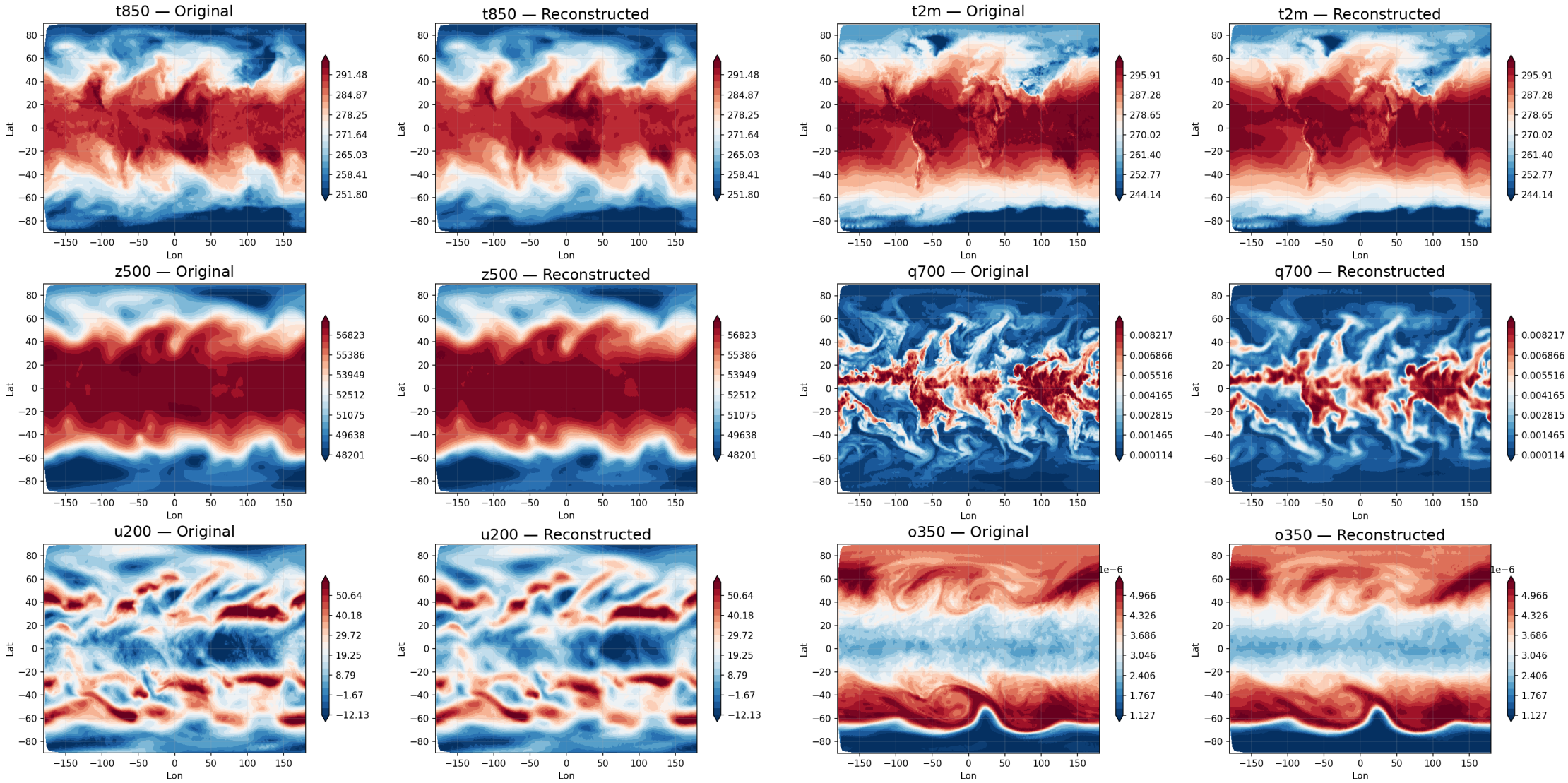
- Experimental setup: 18 pressure levels, N80 resolution
- 6 atmospheric variables (T, u ,v, Z, q, r(O_3)) at each level
- 8 surface variables (sst, t2m, mslp, stl1, siconc, sd) + 6 static features



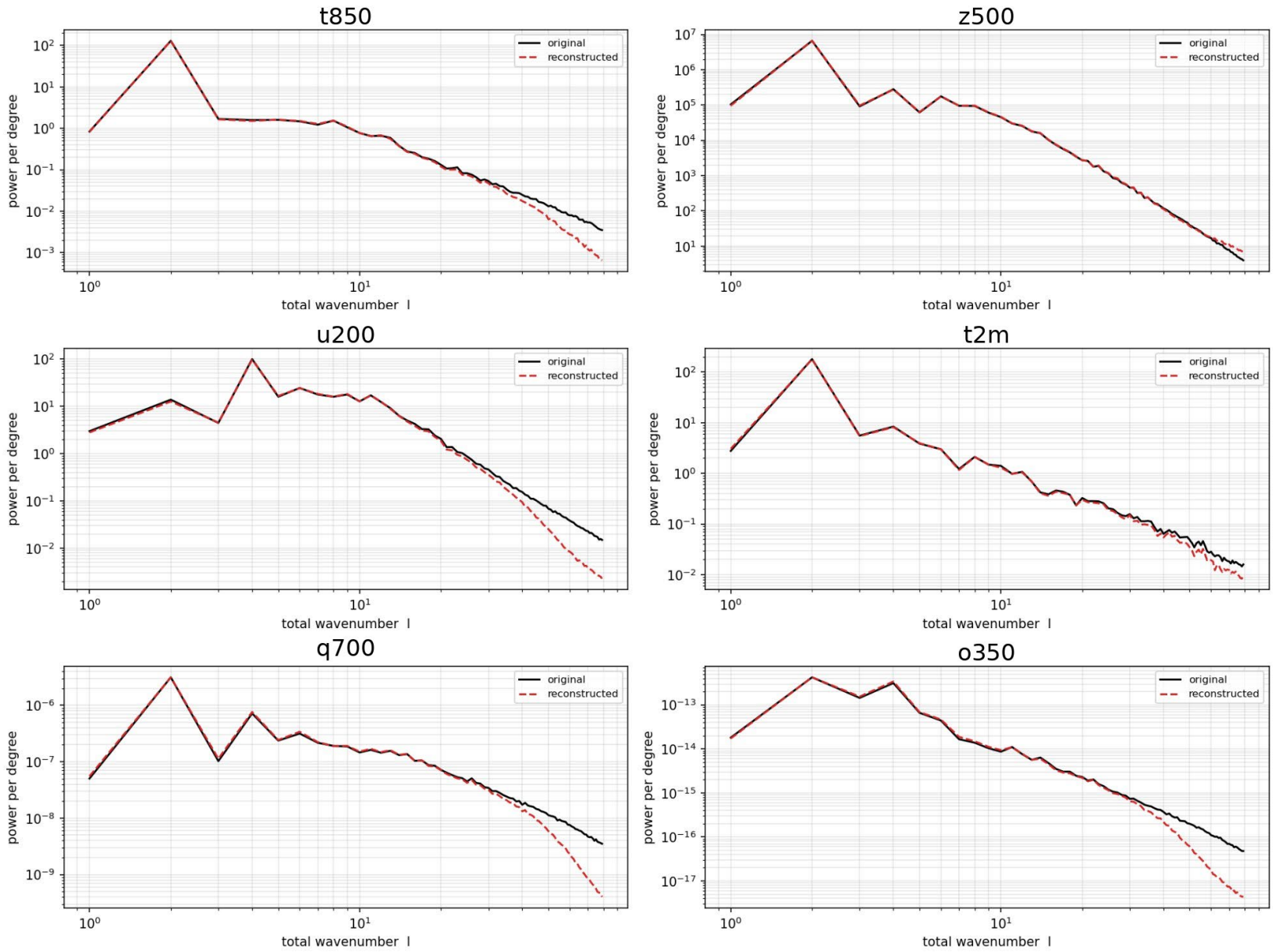
ECMWF

Reconstruction quality

- Some smoothing, due to (1) a10-fold compression at the bottleneck, no information passes it (e.g. through skip connections), (2) the input data is already at large scales only (N80)

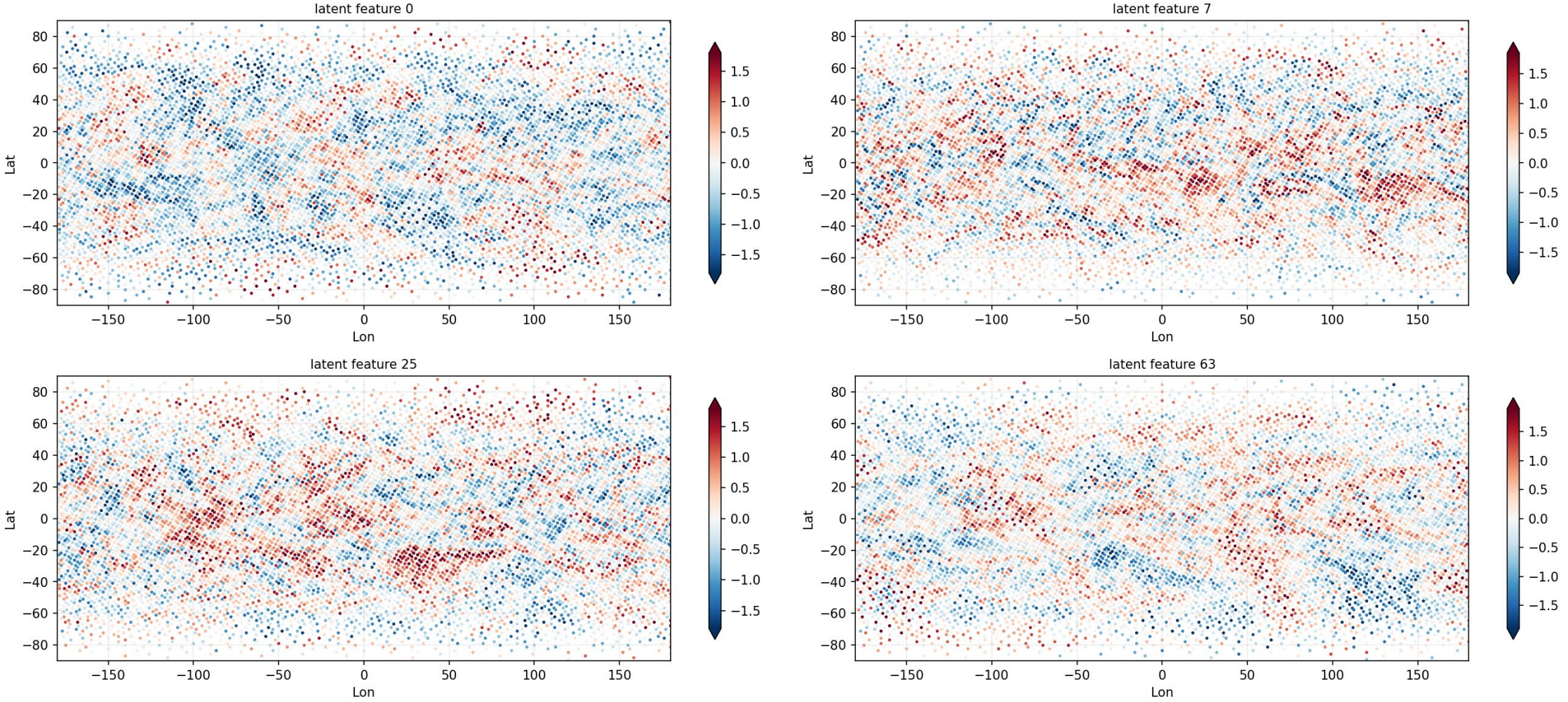


Input and reconstruction spectra

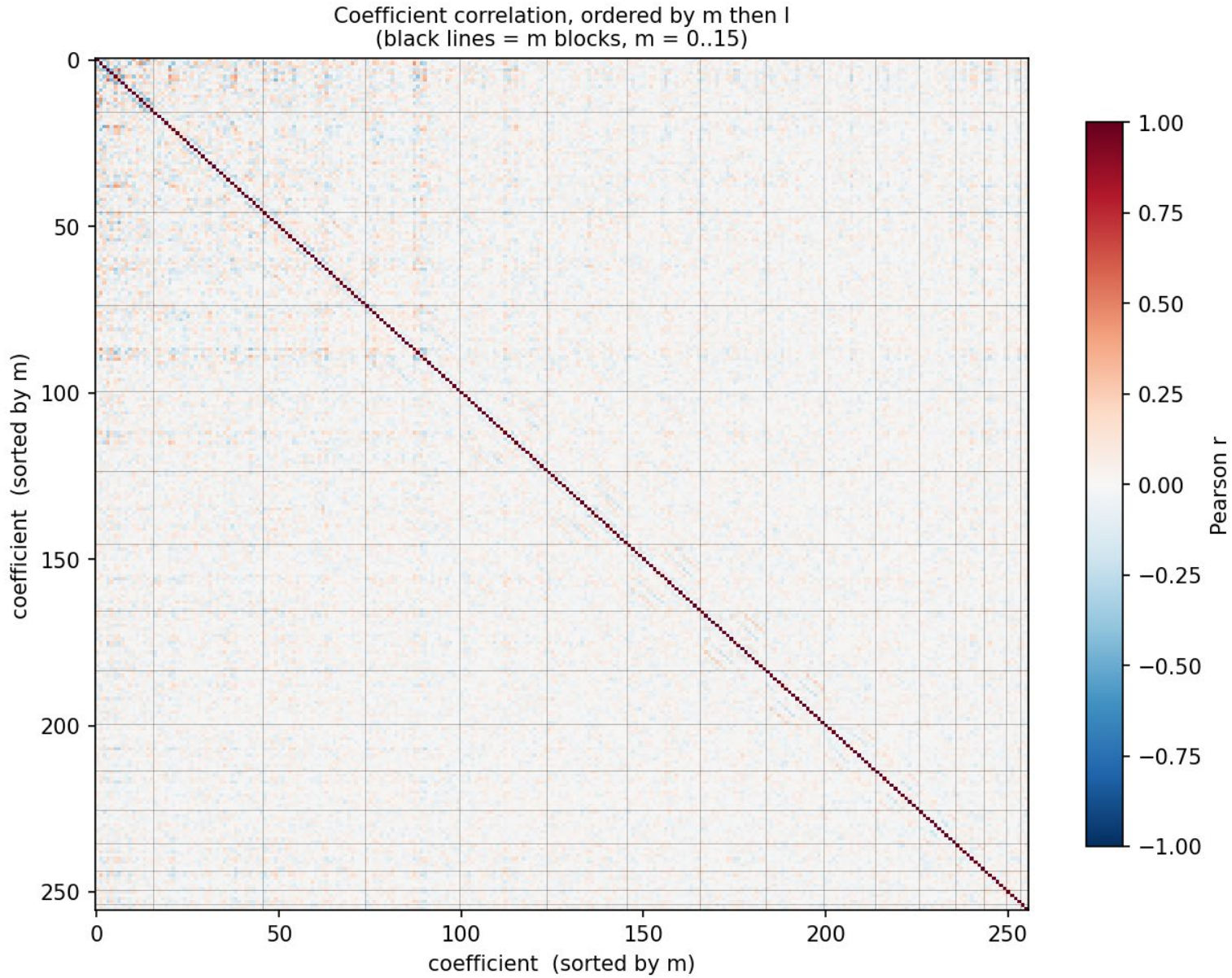


What does the latent space look like?

v108 e155 (best) - latent-space features 2024-06-15 00Z (archive idx 16237)
8929 latent nodes, one dot per node



Are the latent components correlated?



Measuring the B_z-matrix

Encodes EDA perturbed members i = 1,…,9 for every 3 valid dates across 2023–2025. For each date t, nine consecutive circular pairs of perturbed forecasts at 6-hour lead time (initialised at 18 UTC, valid for 00 UTC, on which data the autoencoder is trained)

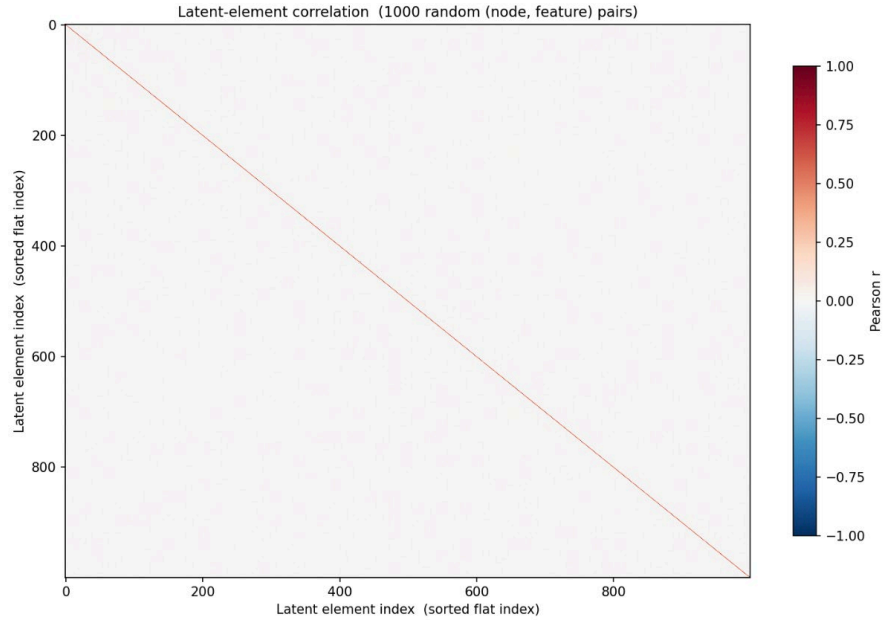
(i,j) ∈ {(1,2), (2,3), …, (8,9), (9,1)} yield latent-space differences:

$$\varepsilon_{ij}(t) = z_{b,i}(t) - z_{b,j}(t)$$

Because both members are drawn from the same background-error distribution N(0,B), the expected outer product is

$$E[\varepsilon_{ij}\ \varepsilon_{ij}{}^{T}] = 2B$$

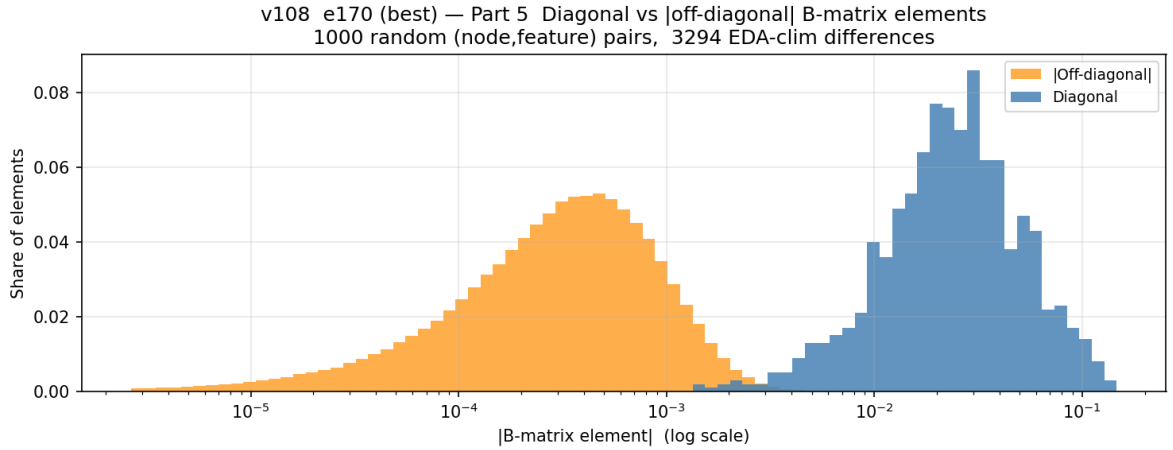
All accumulators are divided by 2. The control (i = 0) was not used in the computations.



Diagonal B-matrix:

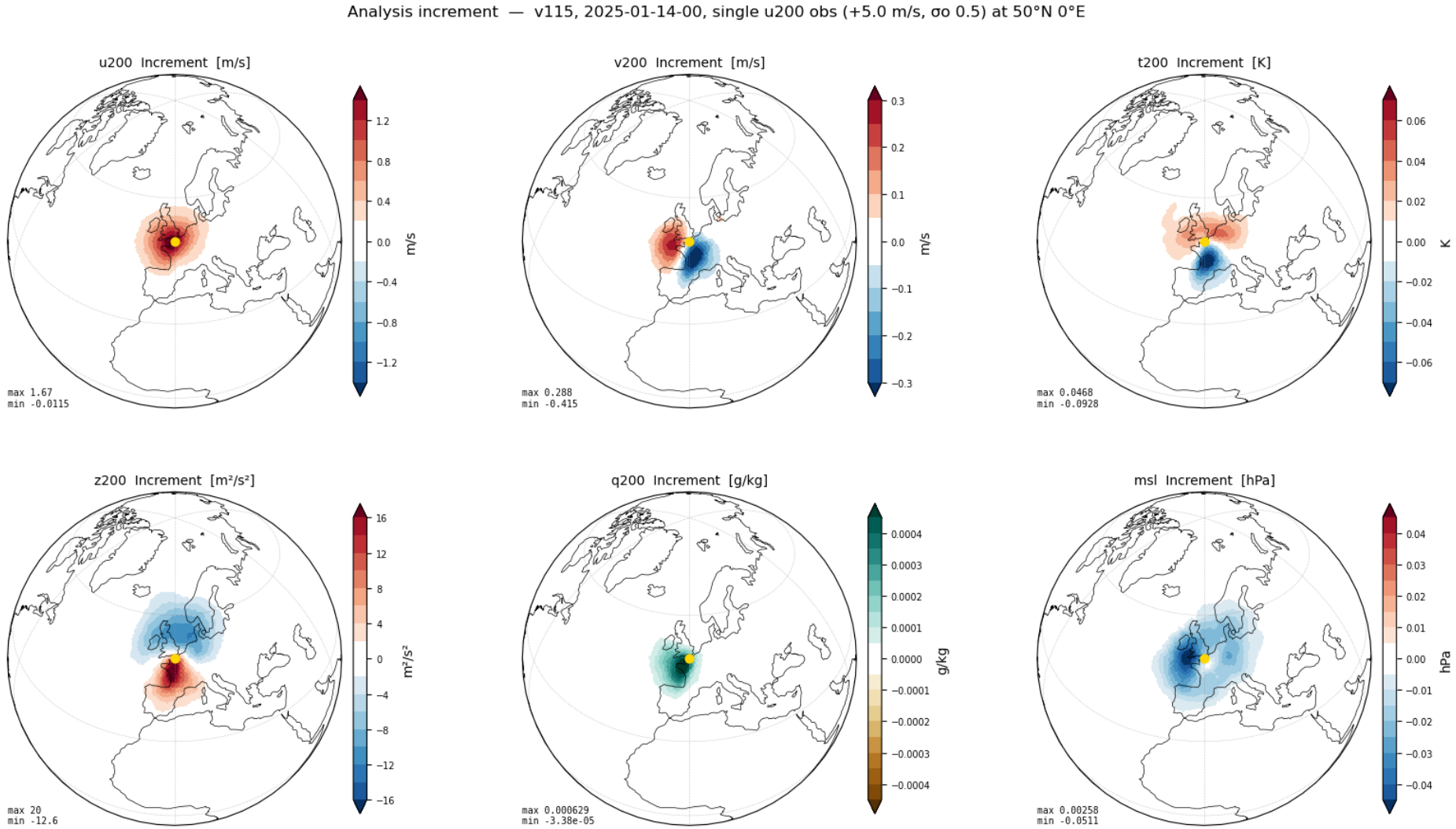
$$B_z^{\text{EDA-clim}}[n,f] = 1/(2T)\ \sum_{t=1,\ldots,T} \sum_{(i,j)}\ \varepsilon_{ij}[n,f](t)^2$$

n… node
f… feature



Single observation experiment

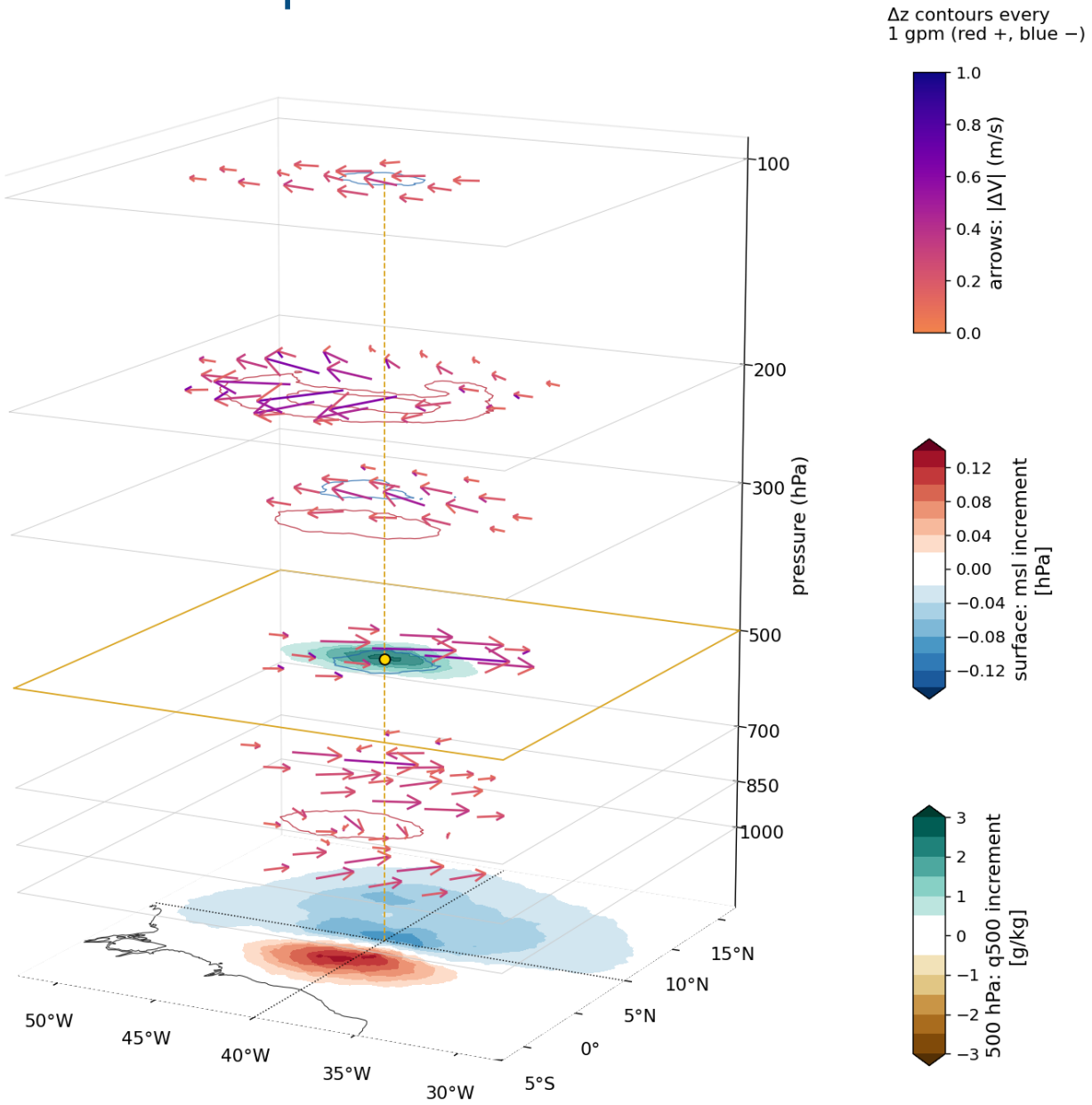
- Observation of zonal wind over London at 200 hPa with departure of 5 m/s and obs. error std. of 0.5 m/s



Single observation experiment - Tropics

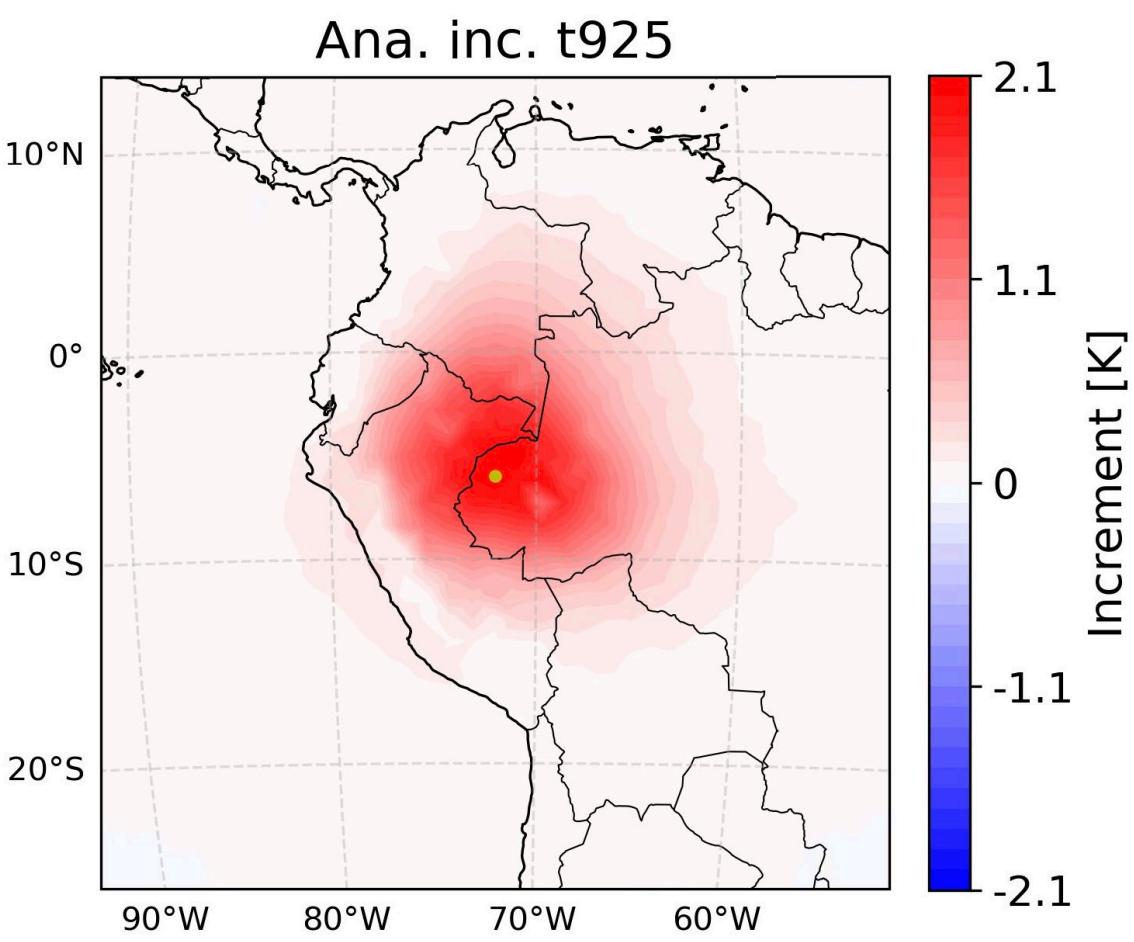
A single observation experiment assimilating synthetic observation of 500 hPa specific humidity with +4g/kg departure on 2024-04-14 at 5N 40W (north of the coast of Brazil):

- Localised horizontally
- Physically-consistent dynamic and thermodynamic response (e.g. pressure-difference driven friction flow at the bottom
- Gill-like response to heat-induced perturbations (upper-divergence and lower-level convergence)



ECMWF

Flow-dependent features: impact of orography



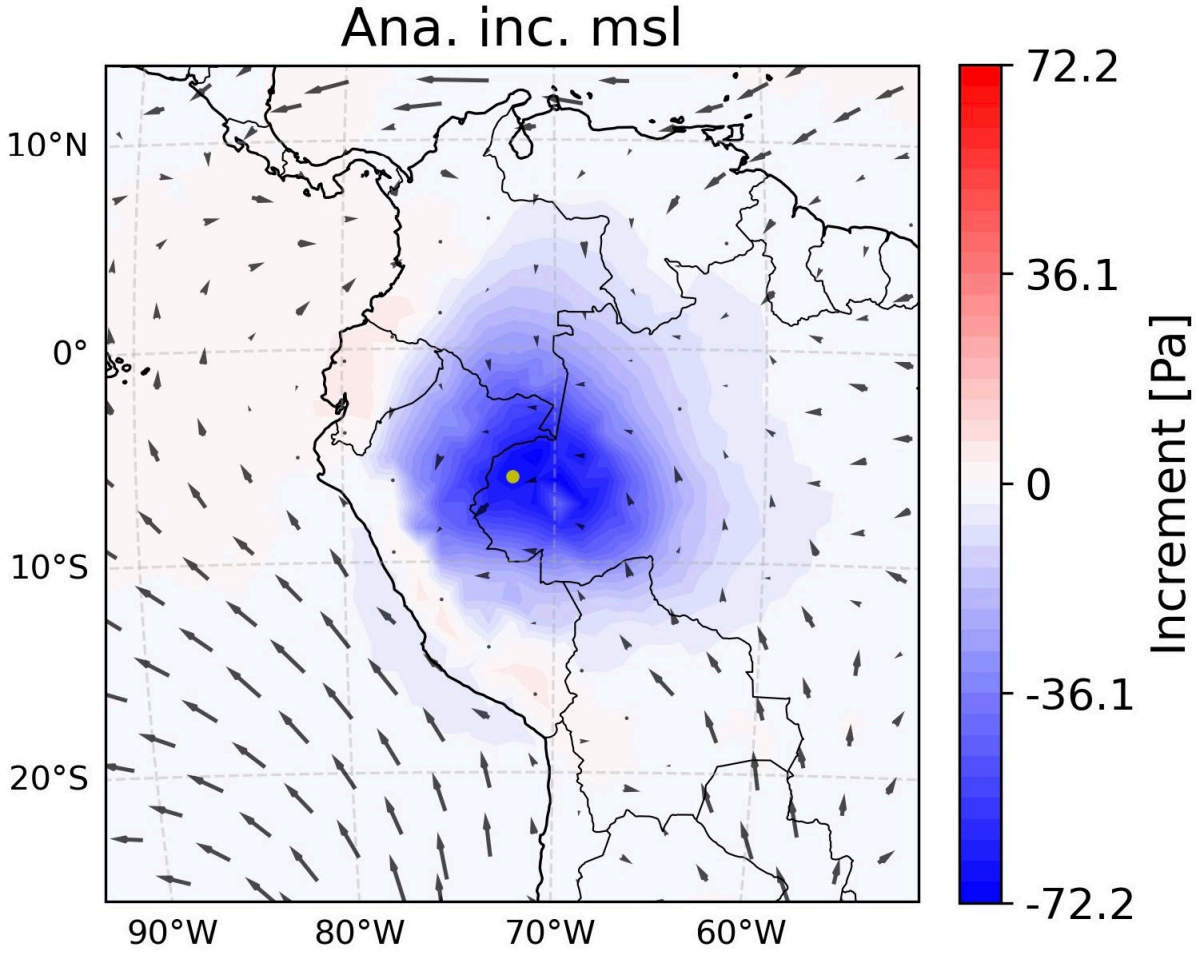
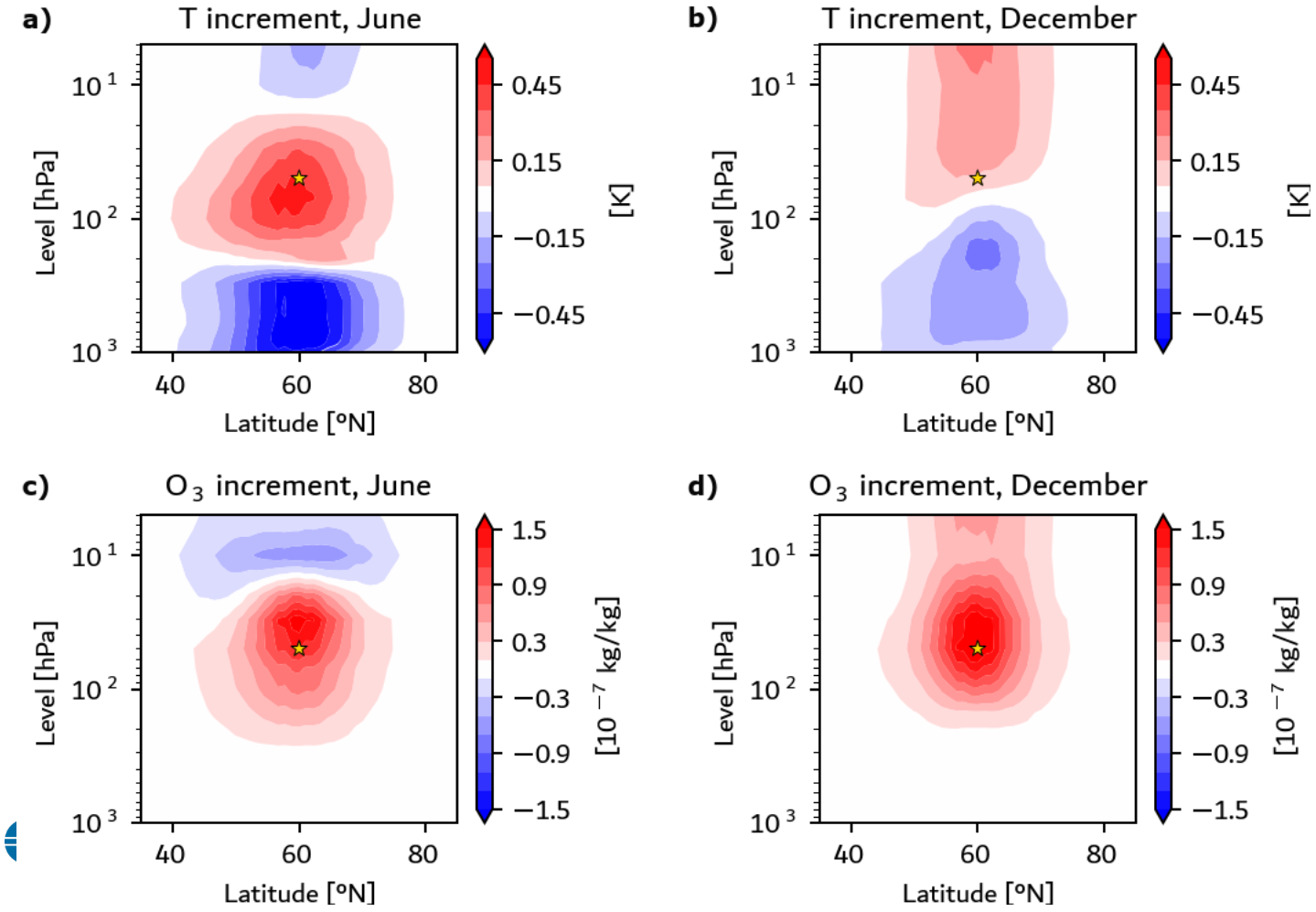


Figure courtesy of Janne Lisa Bouillon
Zaplotnik et al. (2026), in prep.

Stratospheric data assimilation

Operationally, ozone is assimilated univariately, i.e. independently from other atmospheric variables

Single-observation experiments ($d^{O3(50hPa)}$ = +10^{-6} kg/kg) with LS3D-Var using climatological $\mathbf{B}_z$ and 18-level multivariate GNN AE indicate coupled ozone and temperature increments.



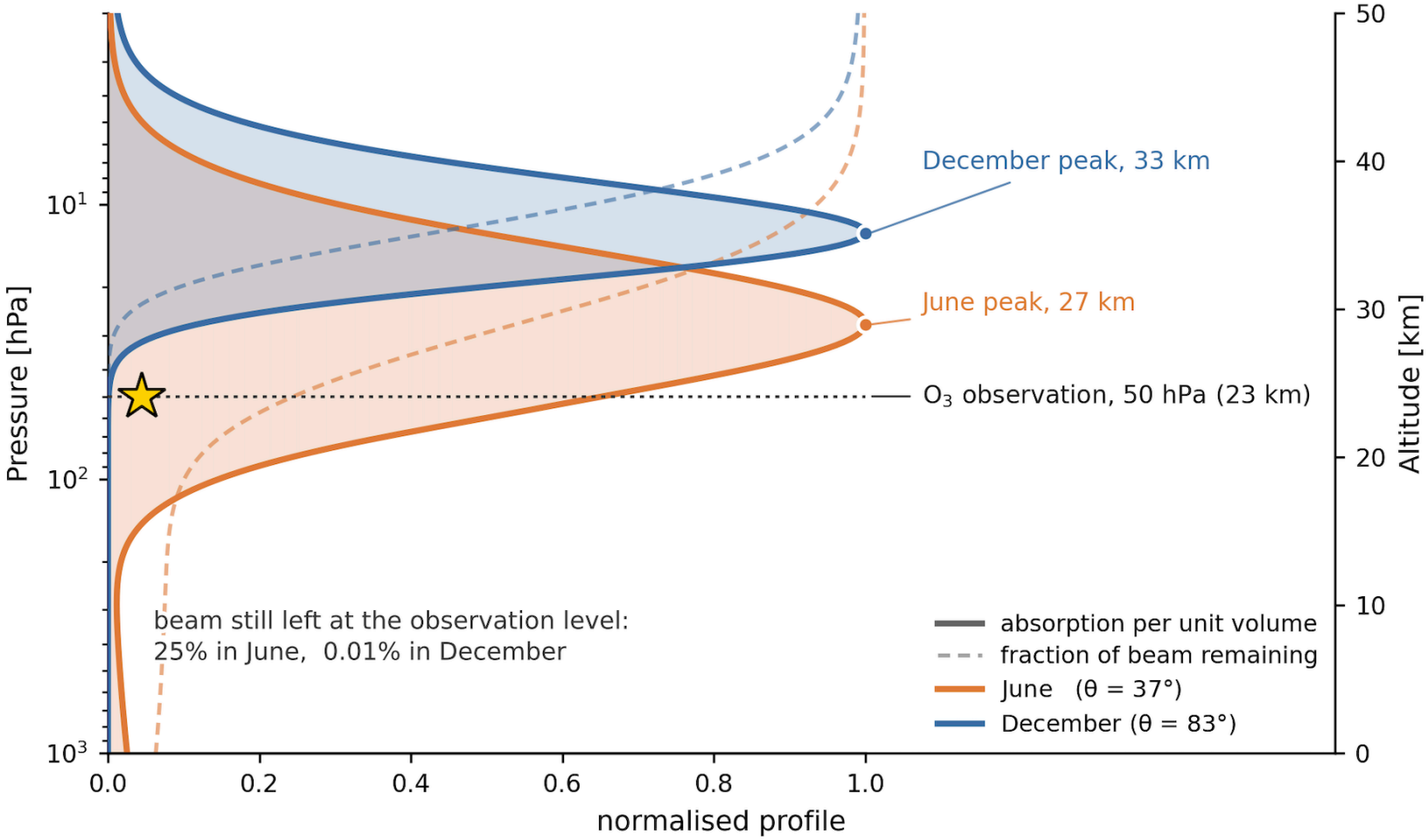
The results show the average increments after performing daily single-obs experiments.

Figure courtesy of Bostjan Melinc Zaplotnik et al. (2026), in prep.

Stratospheric data assimilation

Operationally, ozone is assimilated univariately, i.e. independently from other atmospheric variables

Single-observation experiments ($d^{O3(50hPa)} = +10^{-6}$ kg/kg) with LS3D-Var using climatological $\mathbf{B}_z$ and 18-level multivariate GNN AE indicate coupled ozone and temperature increments



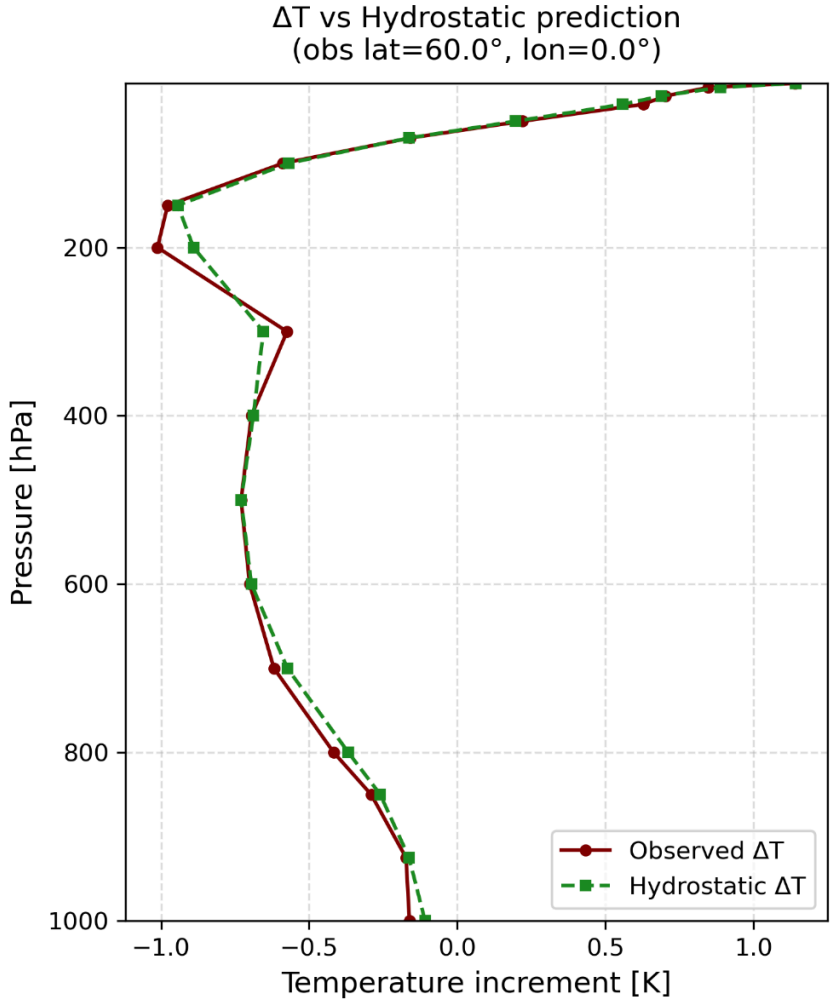
The autoencoder has effectively learned the Beer-Lambert law.

$$\mathrm{d}I = -\mu\, I\, dx$$

Stratospheric data assimilation

In operations, ozone is assimilated univariately, i.e. independently from other atmospheric components

Single-observation experiments ($d^{O3(50hPa)} = +10^{-6}$ kg/kg) with LS3D-Var using climatological $\mathbf{B}_z$ and 18-level multivariate GNN AE indicate coupled ozone and temperature increments



Hydrostatic equilibrium is preserved following assimilation

$$\Delta T \approx -\frac{g}{R_d}\frac{\partial(\Delta Z)}{\partial(\Delta \ln p)}$$

Figure courtesy of Janne Lisa Bouillon Zaplotnik et al. (2026), in prep.

Fully-coupled ocean-atmosphere DA:

- Observation location: (60˚N, 176°W)
- d^{T2m} = -5 K
- Backgrounds with **maximum** sea-ice concentration **(top)** and **minimum** sea-ice concentration **(bottom)**
- Top: positive sea-ice increment along sea-ice boundary
- Bottom: no sea-ice increment!

- **This is the path to fully coupled DA beyond outer loop heuristics**

ECMWF

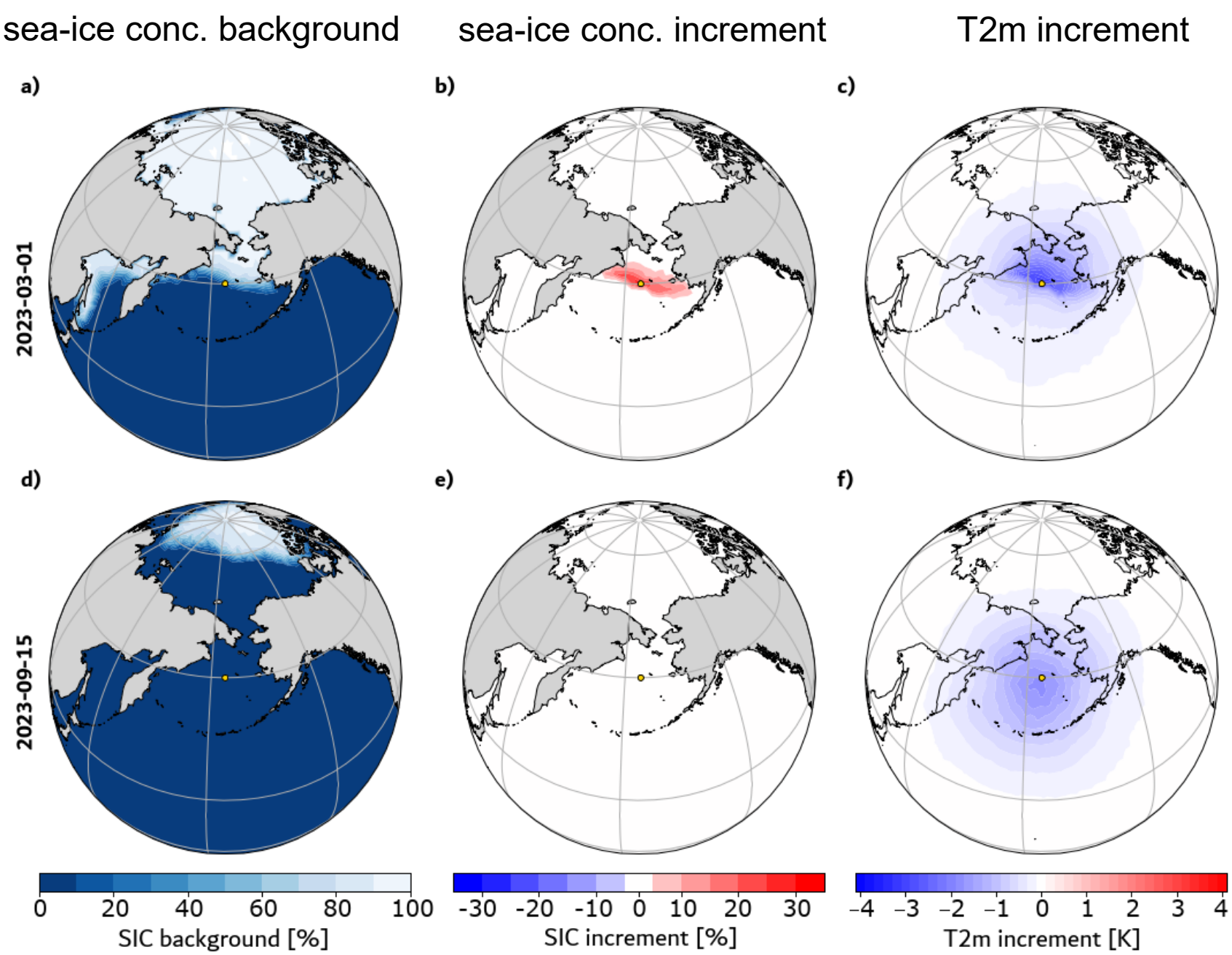
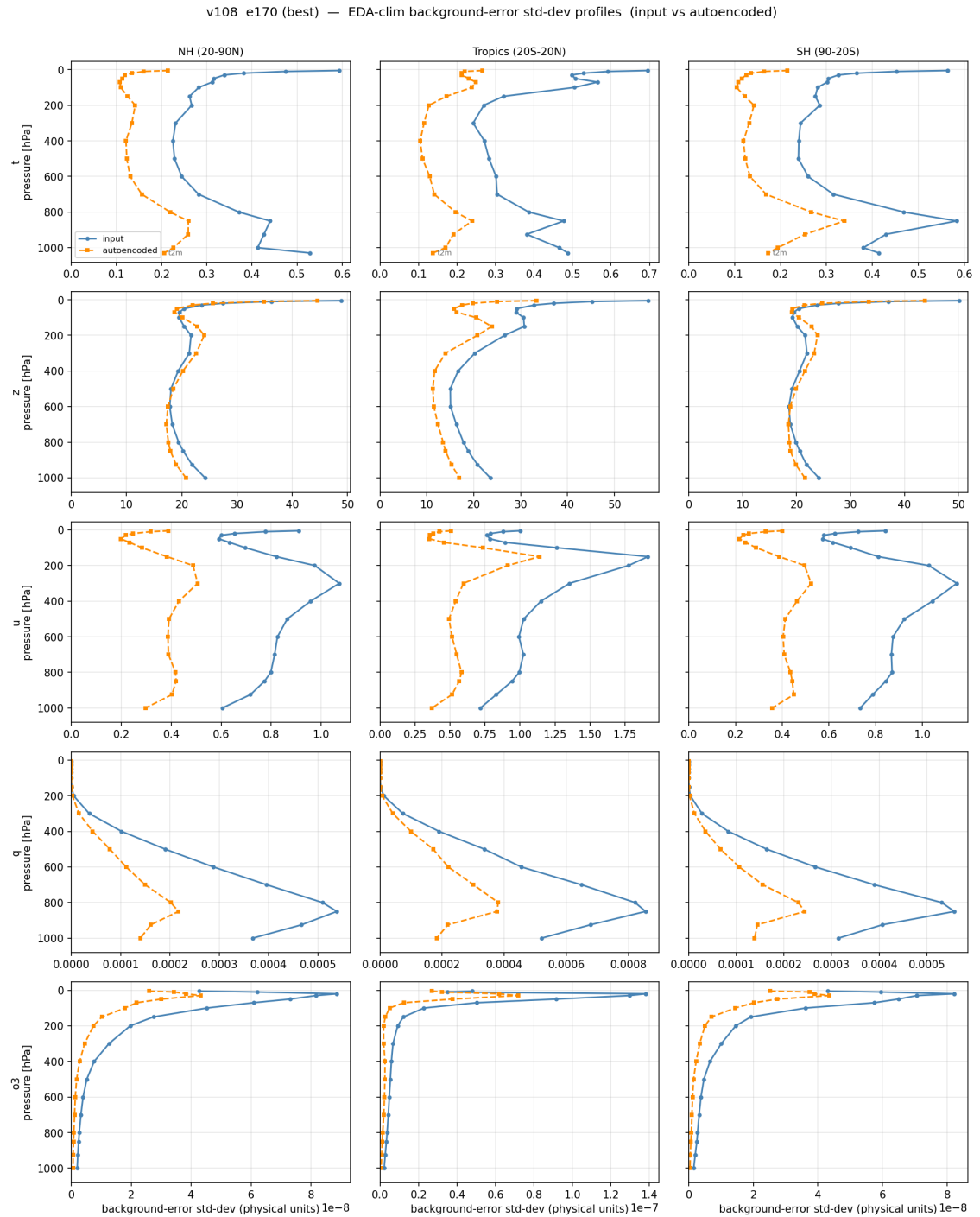


Figure courtesy of Lukas Kugler
Zaplotnik et al. (2026), in prep.

Limitations of N80 autoencoder

EDA variances not yet properly captured, most likely due to insufficient resolution

→ by increasing the resolution of the autoencoder, we're steadily approaching the proper reconstruction



ECMWF

4D-Var latent-space data assimilation

- For the application in IFS/future reanalyses, we will replace the control variable transform with the linearised decoder and its adjoint

- Incremental 4D-Var cost function (only Jo and Jb terms) and its gradient are currently:

$$J(\boldsymbol{\chi}) = \frac{1}{2}(\boldsymbol{\chi}+\boldsymbol{\chi}_g)^T(\boldsymbol{\chi}+\boldsymbol{\chi}_g) + \frac{1}{2}\sum_{k=0}^{N}(\mathbf{d}_k - \mathbf{H}_k\mathbf{M}_{t_0\to t_k}\mathbf{L}\boldsymbol{\chi})^T\mathbf{R}_k^{-1}(\mathbf{d}_k - \mathbf{H}_k\mathbf{M}_{t_0\to t_k}\mathbf{L}\boldsymbol{\chi})$$

$$\nabla_{\boldsymbol{\chi}} J(\boldsymbol{\chi}) = \boldsymbol{\chi}+\boldsymbol{\chi}_g + \sum_{k=0}^{N}\mathbf{L}^T\mathbf{M}_{t_0\to t_k}^T\mathbf{H}_k^T\mathbf{R}_k^{-1}\left(\mathbf{d}_k - \mathbf{H}_k\mathbf{M}_{t_0\to t_k}\mathbf{L}\boldsymbol{\chi}\right)$$

$$\delta\mathbf{x} = \mathbf{L}\boldsymbol{\chi}$$

4D-Var latent-space data assimilation

Square-root preconditioned incremental latent-space 4D-Var becomes:

$$J_{\chi_z}(\boldsymbol{\chi}_z) = \frac{1}{2}(\boldsymbol{\chi}_z+\boldsymbol{\chi}_{zg})^T(\boldsymbol{\chi}_z+\boldsymbol{\chi}_{zg})+\frac{1}{2}\sum_{k=0}^{N}(\mathbf{d}_k-\mathbf{H}_k\mathbf{M}_{t_0\to t_k}\mathbf{S}^{-1}\mathbf{D}\mathbf{L}_z\boldsymbol{\chi}_z)^T\mathbf{R}_k^{-1}(\mathbf{d}_k-\mathbf{H}_k\mathbf{M}_{t_0\to t_k}\mathbf{S}^{-1}\mathbf{D}\mathbf{L}_z\boldsymbol{\chi}_z)$$

$$\nabla_{\boldsymbol{\chi}_z} J(\boldsymbol{\chi}_z) = \boldsymbol{\chi}_z + \boldsymbol{\chi}_{zg} + \sum_{k=0}^{N} \mathbf{L}_z^T\mathbf{D}^T\mathbf{S}^{-1}\mathbf{M}_{t_0\to t_k}^T\mathbf{H}_k^T\mathbf{R}_k^{-1}\left(\mathbf{H}_k\mathbf{M}_{t_0\to t_k}\mathbf{S}^{-1}\mathbf{D}\mathbf{L}_z\boldsymbol{\chi}_z - \mathbf{d}_k\right)$$

- Latent increment $\delta\mathbf{z} = \mathbf{z} - \mathbf{z}_g$
- Guess increment $\delta\mathbf{z}_g = \mathbf{z}_g - \mathbf{z}_b$
- Observation departure $\mathbf{d}_k = \mathbf{y}_k - \mathcal{H}_k\mathcal{M}_{t_0\to t_k}S^{-1}D(\mathbf{z}_g)$
- Control variable transform $\mathbf{L}_z = \mathbf{B}_z^{1/2}$, such that $\delta\mathbf{z} = \mathbf{L}_z\boldsymbol{\chi}_z$

TL approximation: $D(\mathbf{z}) - D(\mathbf{z}_b) = \mathbf{D}\delta\mathbf{z}$

$$\mathbf{D} = (\partial/\partial\mathbf{z})\Big|_{\mathbf{z}=\mathbf{z}_g} D$$

4D-Var latent-space data assimilation – pathway to IFS implementation

1) N128 ML137 pytorch-based autoencoder successfully trained on a single GPU ✓
2) EDA-based climatological B constructed ✓
3) Include torch c++ copy of the pytorch autoencoder in IFS and test in research configuration (work with Marcin and Alban) – **ongoing**

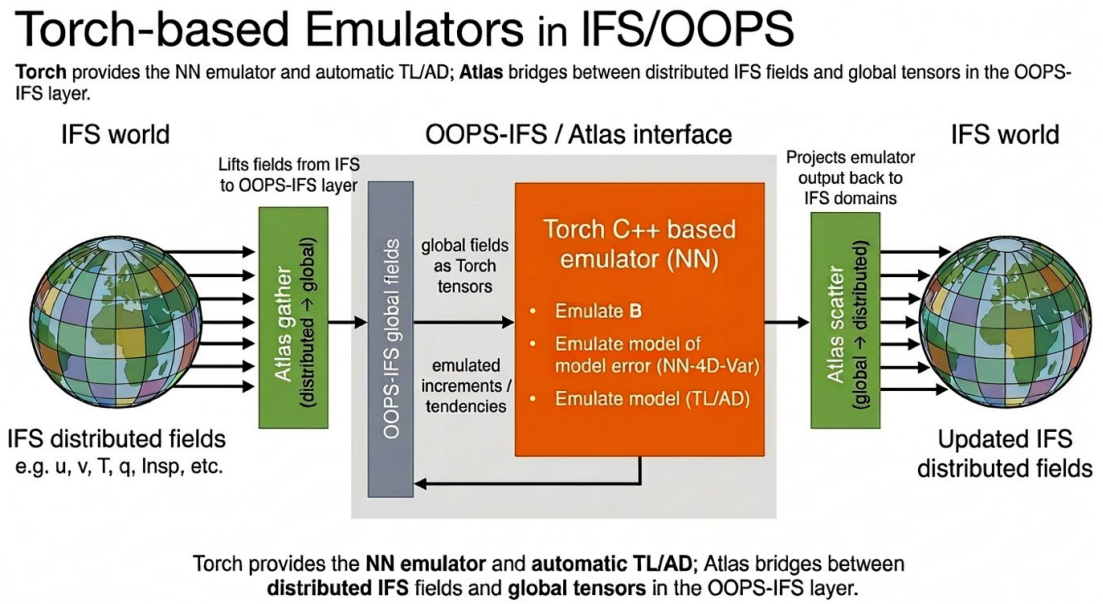


Figure courtesy of Marcin Chrust

See his post on "Supporting NN in OOPS-IFS" https://confluence.ecmwf.int/spaces/ESAS/pages/673320300/DA+Methodology+Team

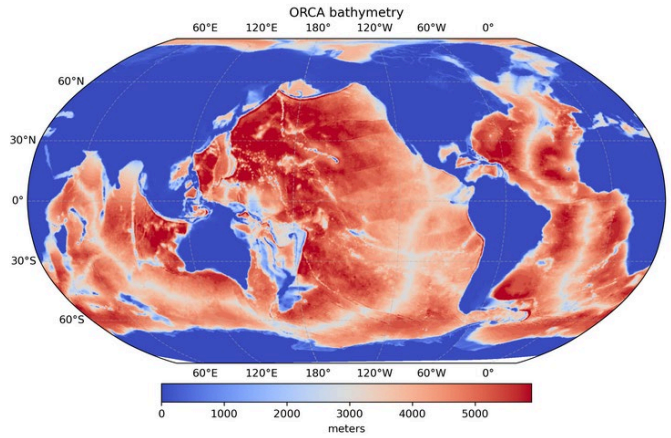
4) N320 ML137 autoencoder (ERA5 data supplemented with ERA6 data) - **ongoing**
5) Dual-resolution autoencoder (trained on ERA5 grid fine-tuned on native ERA6 grid at O800)

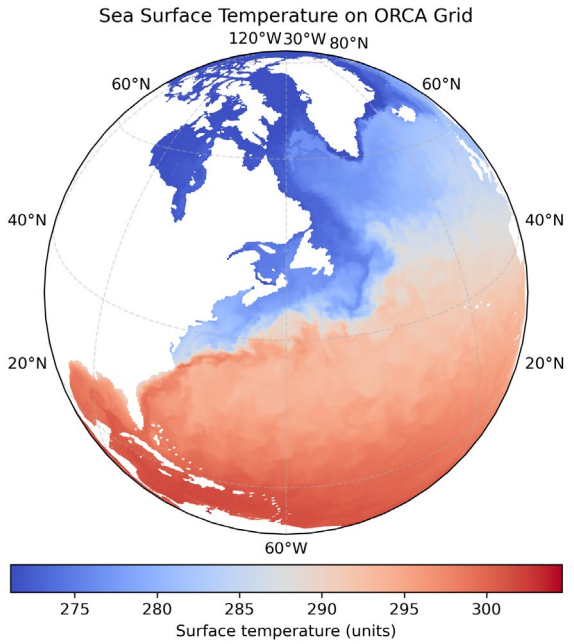
Fully-coupled DA: extension to ocean and land variables

- Preliminary research by Lukas Kugler (University of Ljubljana) – SMASH + preliminary CLIMATE-NEXT activities
- Assimilation in a common latent space of atmosphere, land, ocean, and sea-ice variables

Training data consist of:

- **ERA5**
  - Atmospheric and land variables:
    3-level soil temperature and moisture, snow temperature, snow depth,
    2m temperature, 2m dewpoint
  - Not for the ocean surface (it is a 24h-constant driver in ERA5)
- **ORAS6** (Ocean ReAnalysis 6, unpublished, in verification)
  - 2D: Significant wave height, sea-surface height, sea-ice: area and volume
  - 3D: Temperature, salinity, u/v velocity





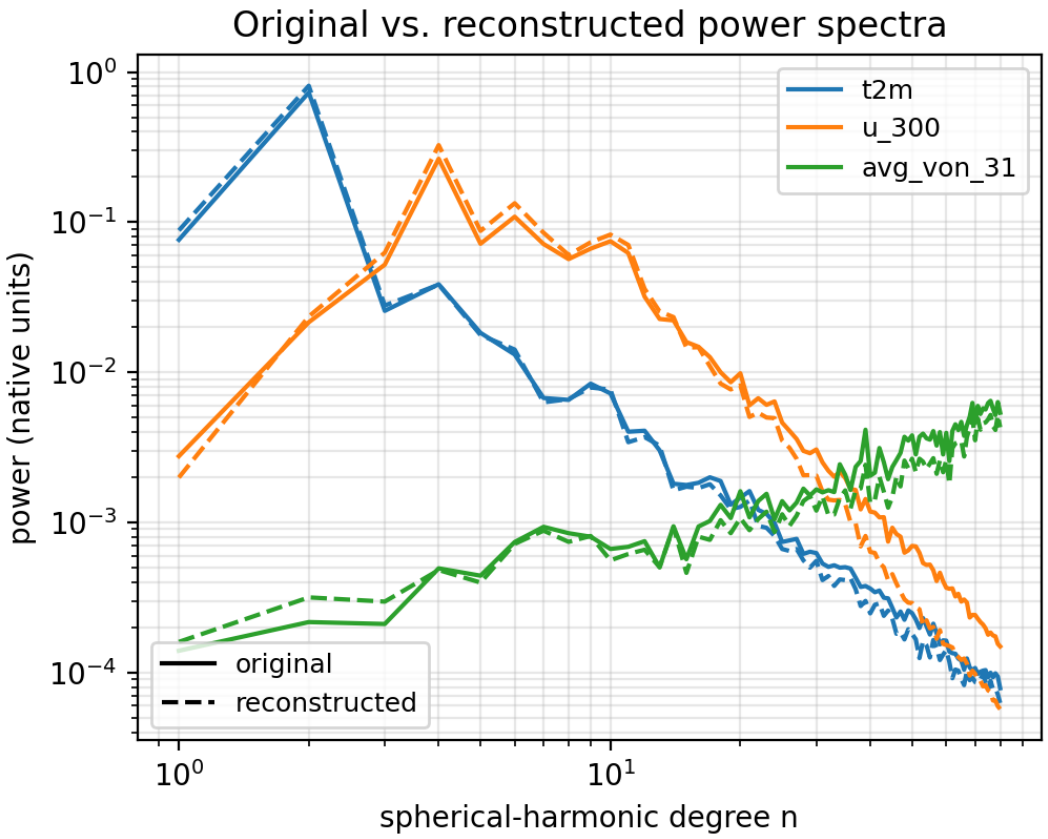
Fully-coupled DA: extension to ocean and land variables

Spatial scales are different:

- In the ocean, mesoscale eddies (~10-100 km) and sub-mesoscale eddies strongly influence forecasts and contain a large portion of the energy
- Thus, getting the small scales right is much more important than in the atmosphere

Time scales are different:

- Ocean variables evolve much slower than atmospheric variables (slower error growth)
- Timescale of mesoscale eddies can exceed weeks



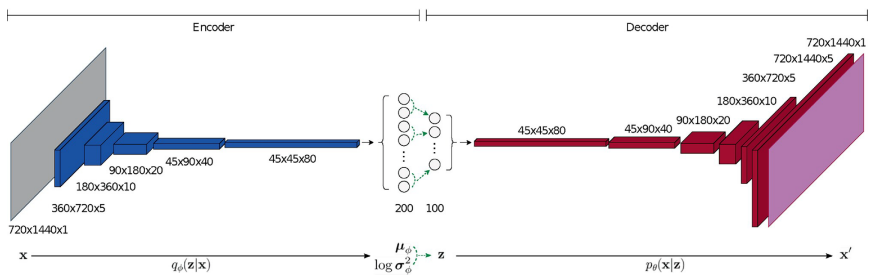
For LSDA, a good autoencoder is all you need… ☺

Requirements of a good autoencoder:

- Performs a significant data compression (O(10x) to O(100x))
- Learns to decorrelate latent-space components (spatially, spectrally, feature-wise)
- Properly reconstructs the initial fields
- Needs some variational flavour

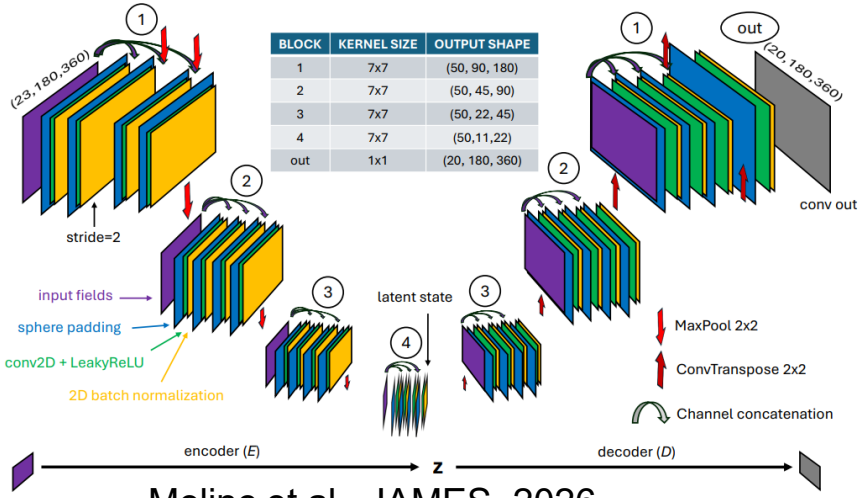
TIMELINE

Proof of concept: simple univariate CNN



Melinc and Zaplotnik, QJRMS, 2024

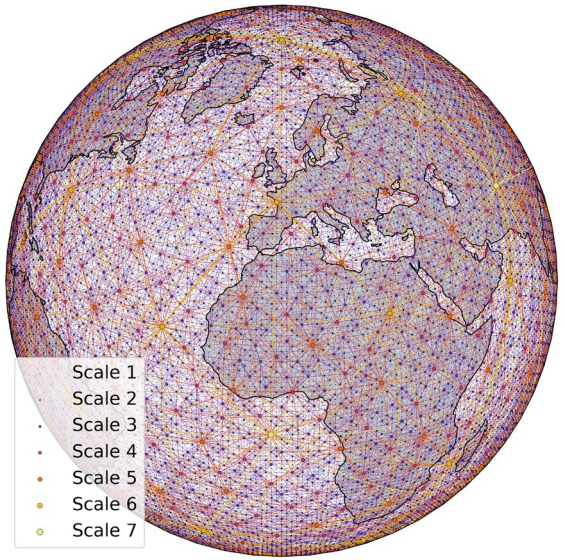
Multivariate multilevel CNN



BLOCK	KERNEL SIZE	OUTPUT SHAPE
1	7x7	(50, 90, 180)
2	7x7	(50, 45, 90)
3	7x7	(50, 22, 45)
4	7x7	(50,11,22)
out	1x1	(20, 180, 360)

Melinc et al., JAMES, 2026

Fully-coupled GNN



Zaplotnik et al., 2026 (in prep.)

Conclusions

Latent space data assimilation:

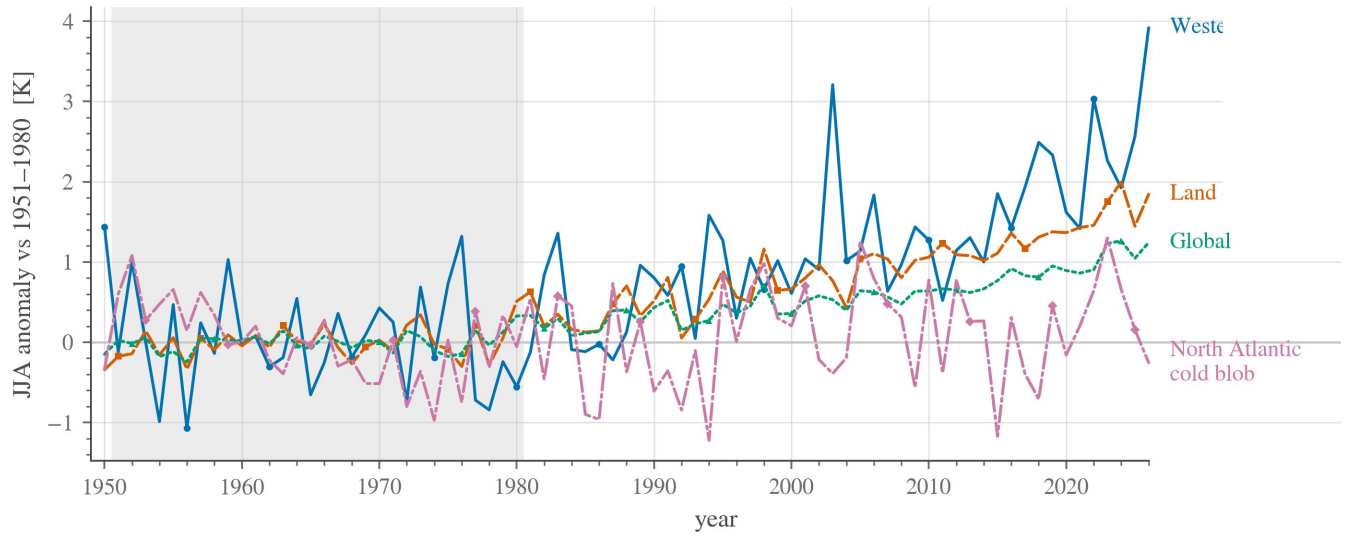
- Produces fully state-dependent analysis increments (flow-dependence can be additionally obtained from errors of the day)
- Analysis increments naturally obey physical boundaries (land-sea boundaries, orography)
- Captures “hidden” atmospheric balances (which would be impossible to be described by general analytical equations in the CVT, e.g. ozone-temperature balances)
- Allows fully-coupled atmosphere-ocean-land DA with a common **B**-matrix
- Yields equivalent solution to standard variational DA (if the decoder is affine)
- EDA variances can be captured with an autoencoder of proper resolution
- **Suitable for incremental 4D-Var in IFS: current effort**

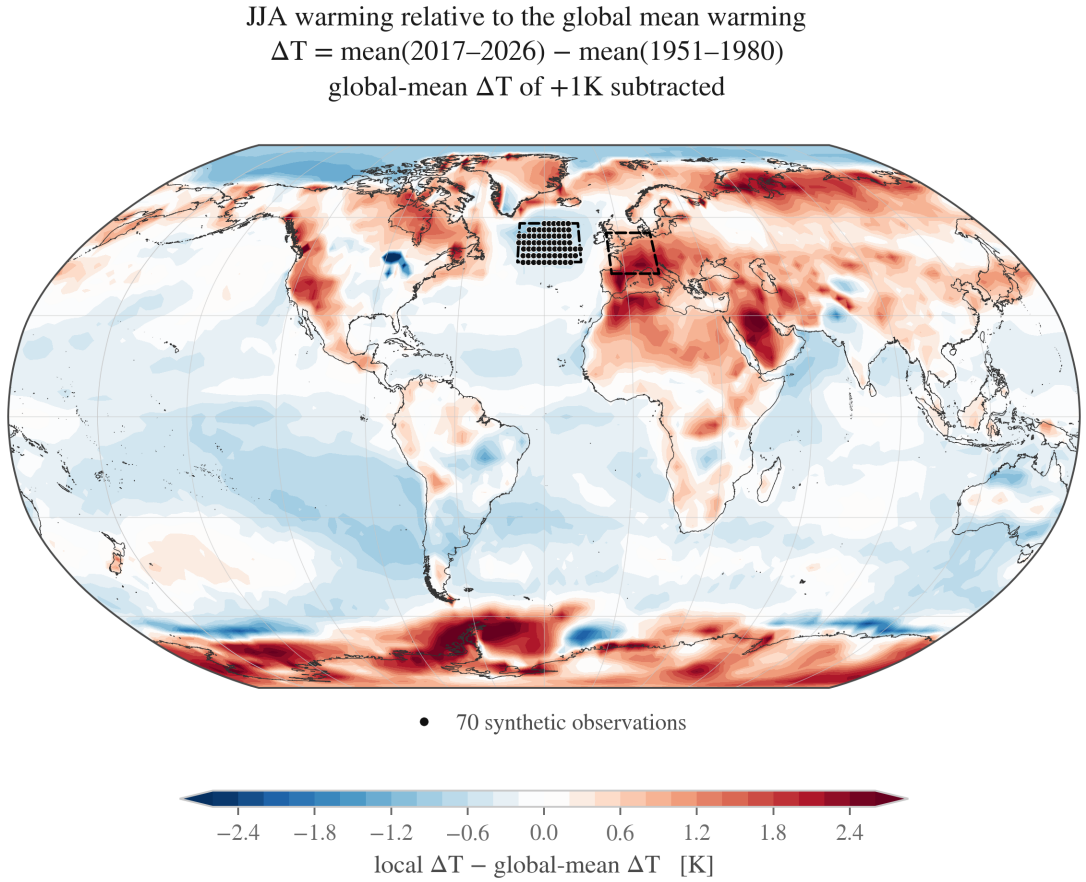
Did you enjoy the weather last summer?

Western Europe is one of the fastest warming areas on the globe, particularly in the summer months.

This cannot be just a normal global warming trend but must be associated with changes in atmospheric circulation.

What does it have to do with DA?





Zaplotnik (2026), European heatwaves are intensified by a cool ocean, in prep.

Did you enjoy the weather last summer?

- When paired with the climatological covariances, the autoencoder acts as a generalised descriptor of the atmospheric dynamics

- How does a perturbation of certain atmospheric/land/sea variable affect other fields? → AE can answer

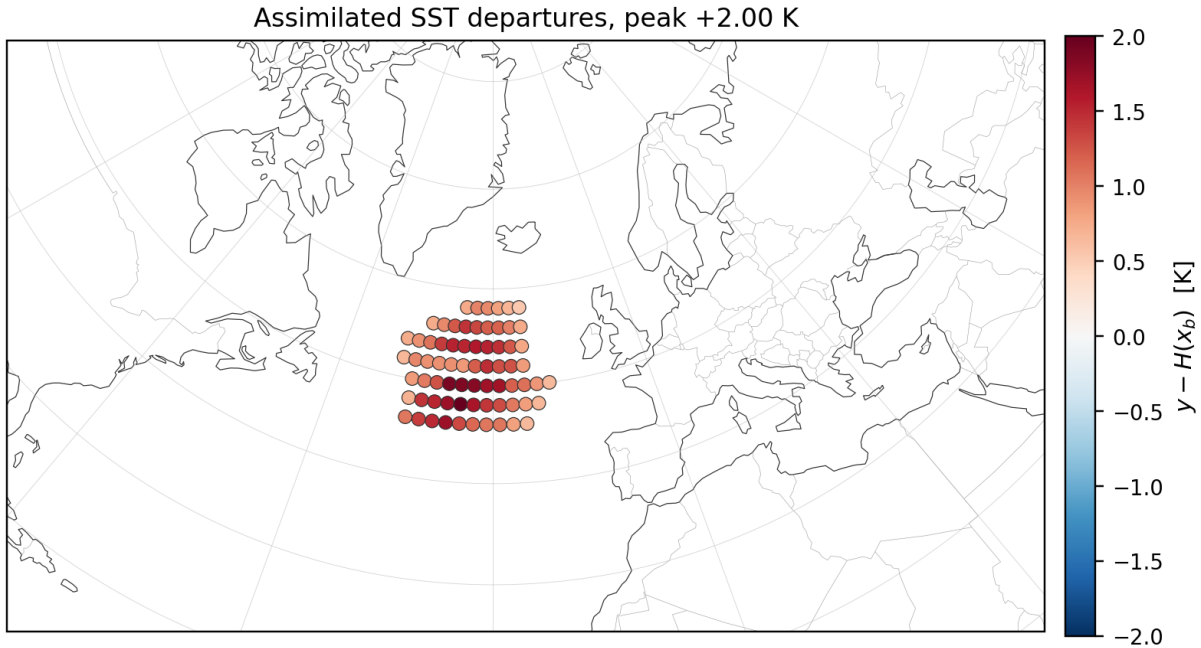
Procedure:

1) represent the ensemble of JJA climatological anomalies in latent space → pack the covariances into $\mathbf{B}_z$

2) Create synthetic observations, which offset the North Atlantic Cold Blob (the “warming hole”)

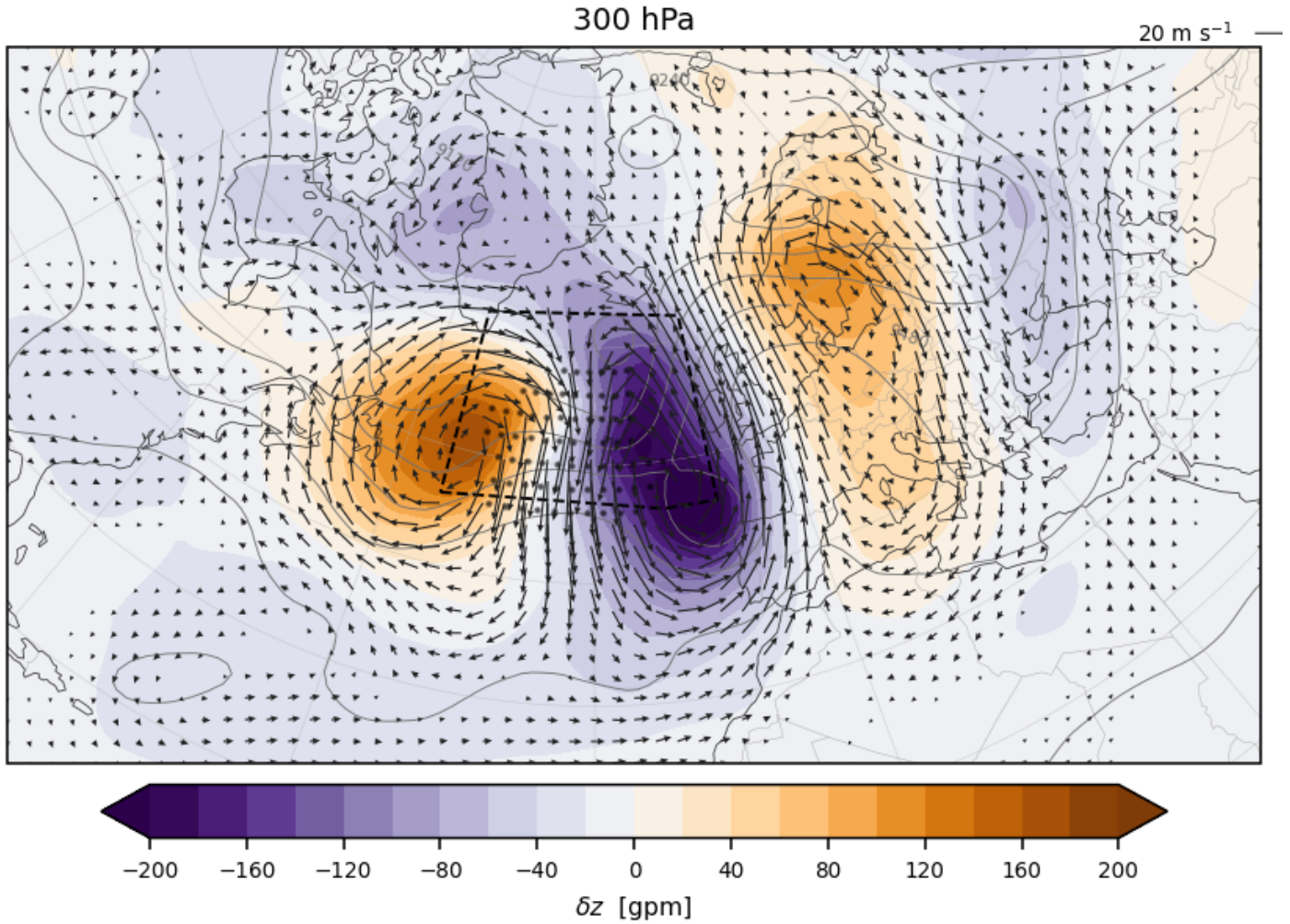
3) Perform 3D-Var in latent space

4) The negative of the response represents the impact of the cold blob on the European summers



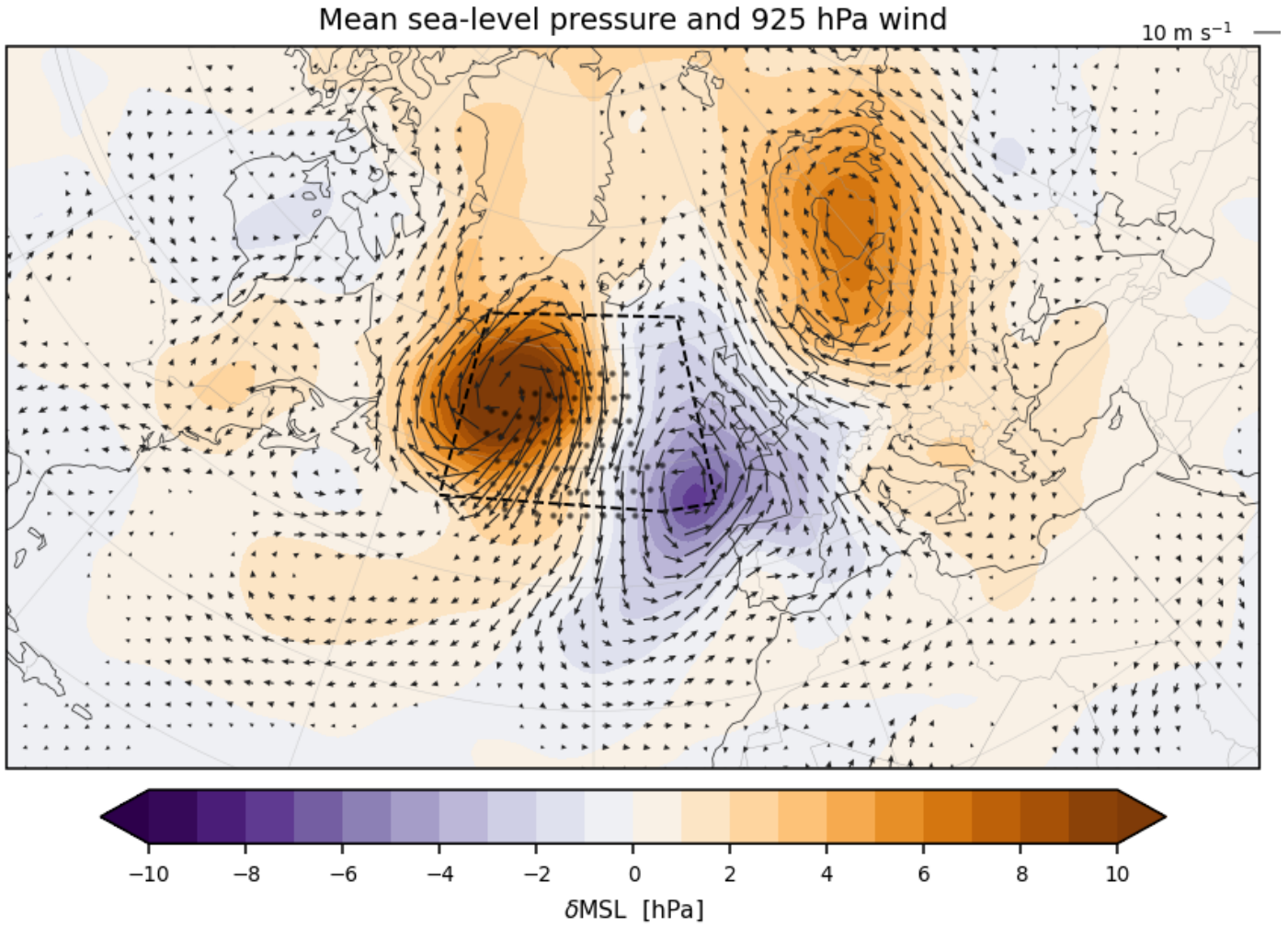
Zaplotnik (2026), European heatwaves are intensified by a cool ocean, in prep.

Did you enjoy the weather last summer?



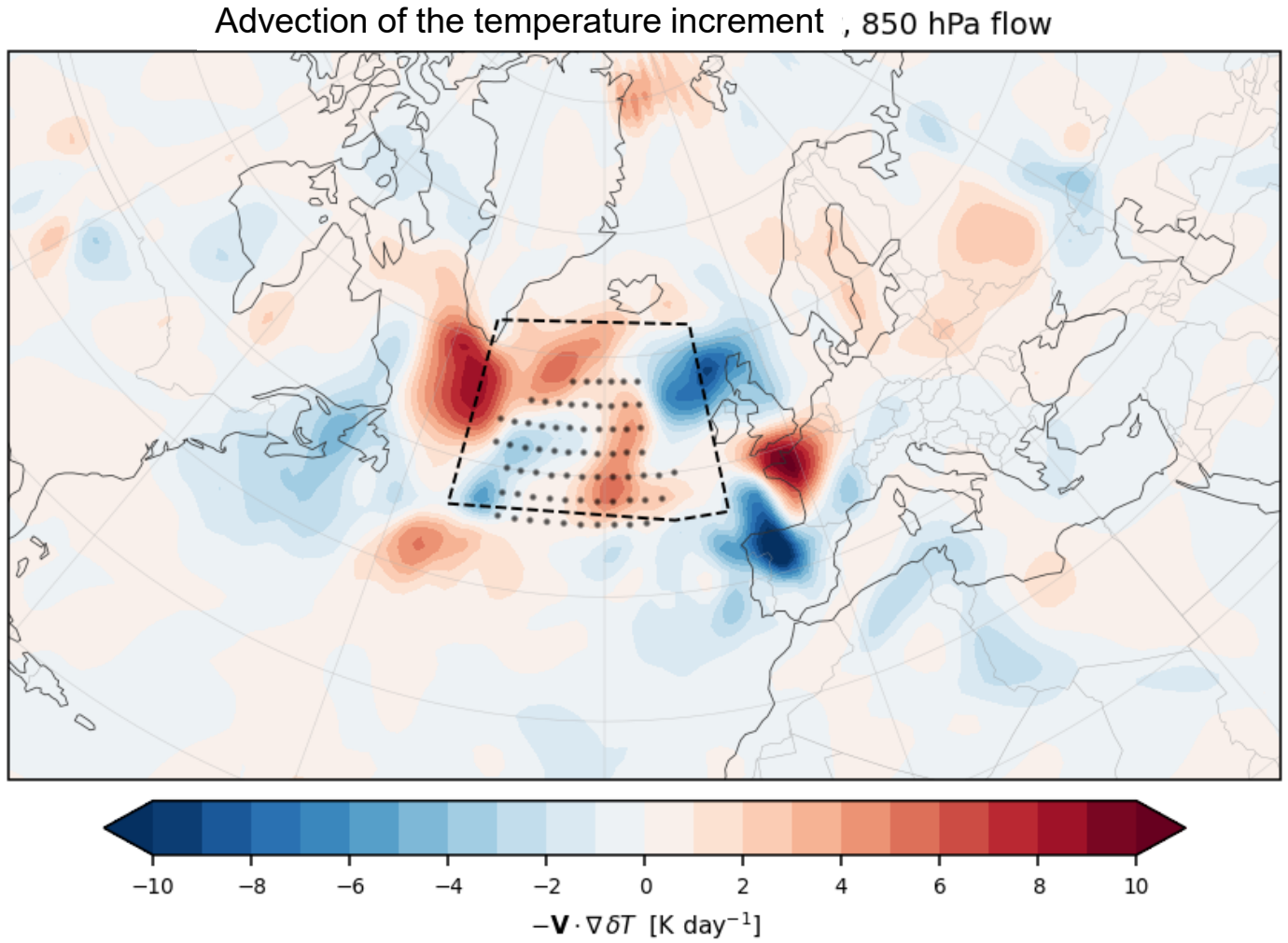
Zaplotnik (2026), European heatwaves are intensified by a cool ocean, in prep.

Did you enjoy the weather last summer?



Zaplotnik (2026), European heatwaves are intensified by a cool ocean, in prep.

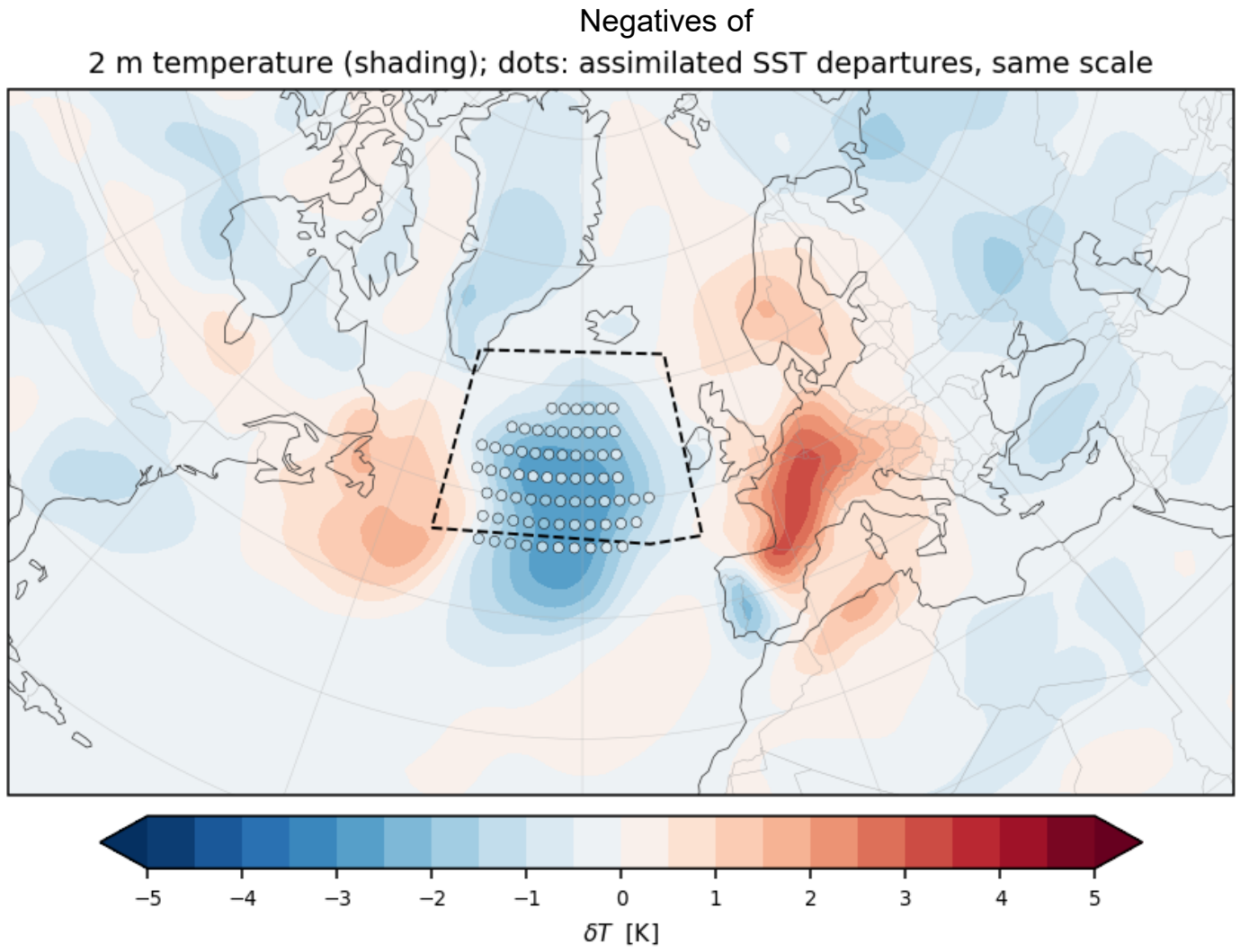
Did you enjoy the weather last summer?



Zaplotnik (2026), European heatwaves are intensified by a cool ocean, in prep.

Did you enjoy the weather last summer?

If not, blame the cold blob – a consequence of warming planet itself - which is intensifying the European heatwaves.



Zaplotnik (2026), European heatwaves are intensified by a cool ocean, in prep.

References

H. Fan, L. Bai, B. Fei, Y. Xiao, K. Chen, Y. Liu, Y. Qu, F. Ling, and P. Gentine. *Physically consistent global atmospheric data assimilation with machine learning in latent space.* Science Advances, 12(1):eaea4248, 2026a. doi: 10.1126/sciadv.aea4248.

H. Fan, J. Nathaniel, Y. Xiao, C. Bian, F. Ling, B. Fei, L. Bai, and P. Gentine. *Accurate and Efficient Hybrid-Ensemble Atmospheric Data Assimilation in Latent Space with Uncertainty Quantification.* Arxiv preprint, 2026b. doi:10.48550/arXiv.2603.04395.

A. E. Gill. Some simple solutions for heat-induced tropical circulation. Quarterly Journal of the Royal Meteorological Society, 106(449):447–462, 1980. doi: 10.1002/qj.49710644905

B. Melinc and Z. Zaplotnik. *3D-Var data assimilation using a variational autoencoder.* Quarterly Journal of the Royal Meteorological Society, 150(761):2273–2295, 2024. doi: 10.1002/qj.4708.

B. Melinc, U. Perkan, and Z. Zaplotnik. *A Unified Neural Background-Error Covariance Model for Midlatitude and Tropical Atmospheric Data Assimilation.* Journal of Advances in Modeling Earth Systems, 18(1):e2025MS005360, 2026. doi: 10.1029/2025MS005360.

I. Pasmans, Y. Chen, T. Sebastian Finn, M. Bocquet, and A. Carrassi. *Ensemble Kalman filter in latent space using a variational autoencoder pair.* Quarterly Journal of the Royal Meteorological Society, 2025. doi: 10.1002/qj.70070.

M. Peyron, A. Fillion, S. Gürol, V. Marchais, S. Gratton, P. Boudier, and G. Goret. *Latent space data assimilation by using deep learning.* Quarterly Journal of the Royal Meteorological Society, 147(740):3759–3777, 2021. doi: 10.1002/qj.4153.

References

M. Amendola, R. Arcucci, L. Mottet, C. Q. Casas, S. Fan, C. Pain, P. Linden, and Y. K. Guo. *Data Assimilation in the Latent Space of a Convolutional Autoencoder.* In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 12746 LNCS, pages 373–386. Springer Science and Business Media Deutschland GmbH, 2021. doi: 10.1007/978-3-030-77977-1_30.

I. Buchnik, D. Steger, G. Revach, R. J. G. van Sloun, T. Routtenberg, and N. Shlezinger. *Learned Kalman Filtering in Latent Space with High-Dimensional Data.* In ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023. doi: 10.1109/ICASSP49357.2023.10096006.

M. K. Davey and A. E. Gill. *Experiments On Tropical Circulation With A Simple Moist Model*. Quarterly Journal of the Royal Meteorological Society, 113(478):1237–1269, 1987. doi: 10.1002/qj.49711347809.

S. A. Falconer, D. J. Lloyd, and N. Santitissadeekorn. *Tracking and forecasting oscillatory data streams using Koopman autoencoders and Kalman filtering.* Physica D: Nonlinear Phenomena, 476:134649, 2025. doi:10.1016/j.physd.2025.134649.

H. Fan, Y. Liu, Z. Huo, Y. Liu, Y. Shi, and Y. Li. *A Novel Latent Space Data Assimilation Framework with Autoencoder-Observation to Latent Space (AE-O2L) Network. Part I: The Observation-Only Analysis Method.* Monthly Weather Review, 153(8):1335–1348, 2025a. doi: 10.1175/MWR-D-24-0057.1.

H. Fan, Y. Liu, Y. Liu, Z. Huo, B. Chen, and Y. Qin. *A Novel Latent Space Data Assimilation Framework with Autoencoder-Observation to Latent Space (AE-O2L) Network. Part II: Observation and Background Assimilation with Interpretability.* Monthly Weather Review, 153(8):1349–1363, 2025b. doi: 10.1175/MWR-D-24-0058.1.

References

S. Cheng, J. Chen, C. Anastasiou, P. Angeli, O. K. Matar, Y.-K. Guo, C. C. Pain, and R. Arcucci. *Generalised Latent Assimilation in Heterogeneous Reduced Spaces with Machine Learning Surrogate Models.* Journal of Scientific Computing, 94(1):11, 2023. doi: 10.1007/s10915-022-02059-4.

S. Cheng, Y. Zhuang, L. Kahouadji, C. Liu, J. Chen, O. K. Matar, and R. Arcucci. *Multi-domain encoder–decoder neural networks for latent data assimilation in dynamical systems.* Computer Methods in Applied Mechanics and Engineering, 430:117201, 2024. doi: 10.1016/j.cma.2024.117201.

K. Wang, S. Cheng, M. D. Piggott, S. L. Dance, Y. Wang, and R. Arcucci. *Latent data assimilation with non-explicit observation operator in hydrology*. Quarterly Journal of the Royal Meteorological Society, 151(772):e5009, 2025. doi: 10.1002/qj.5009.

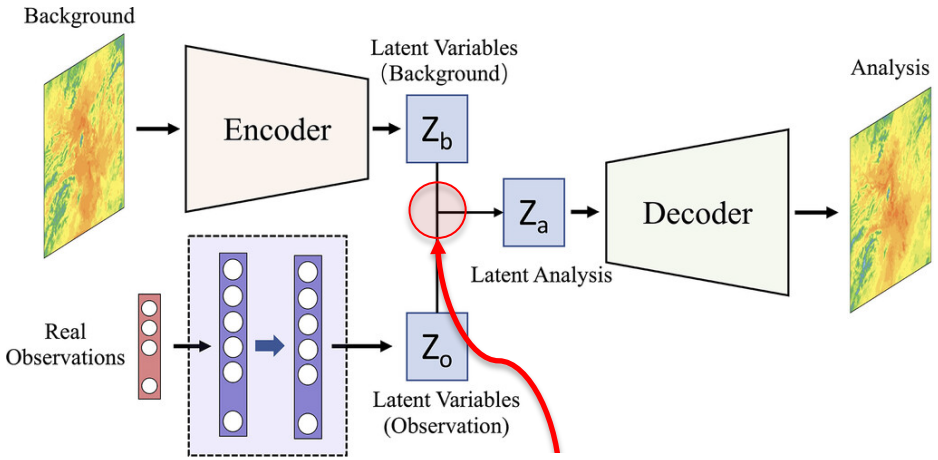
Q. Zheng, Q. Shao, G. Han, W. Li, H. Li, and X. Wang. *Generating unseen nonlinear evolution in the ocean using deep learning-based latent space data* assimilation model. Ocean Modelling, 200:102677, 2026. doi:10.1016/j.ocemod.2026.102677.

Zaplotnik Ž, Žagar N, Gustafsson N. An intermediate-complexity model for four-dimensional variational data assimilation including moist processes. Q J R Meteorol Soc. 2018;144:1772–1787. https://doi.org/10.1002/qj.3338

What can we learn from latent-space DA approaches so far?

- Fan et al. (2025, MWR) compared
  - the observation-only analyses (OOA) vs.
  - the observation+background analyses (OBA), i.e. analyses obtained by merging the information from background and real observations
- Conclusion: the background (~ the model) is crucial, as long as we don't have very densely observed atmosphere

Initialising from observations+background encoded to shared latent space is better.



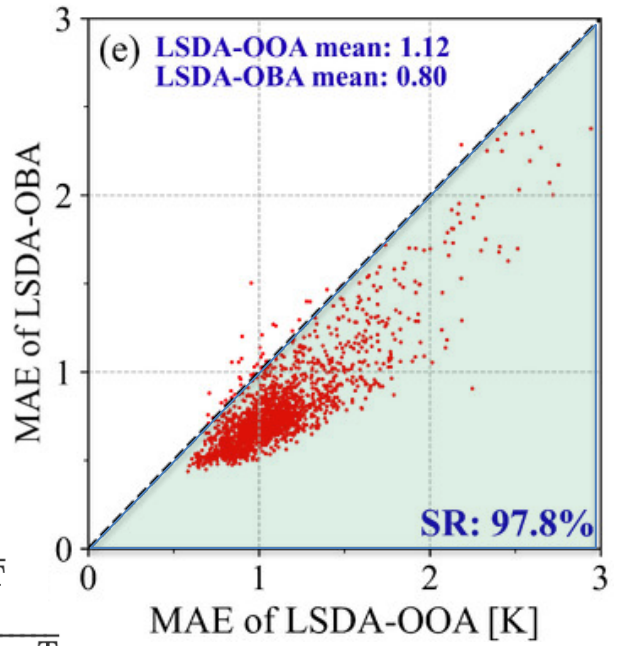
$$\mathbf{z}_a = \mathbf{z}_b + \mathbf{K}_z\left(\mathbf{z}_o - \mathbf{z}_b\right)$$

$$\mathbf{K}_z = \mathbf{B}_z(\mathbf{B}_z + \mathbf{R}_z)^{-1}$$

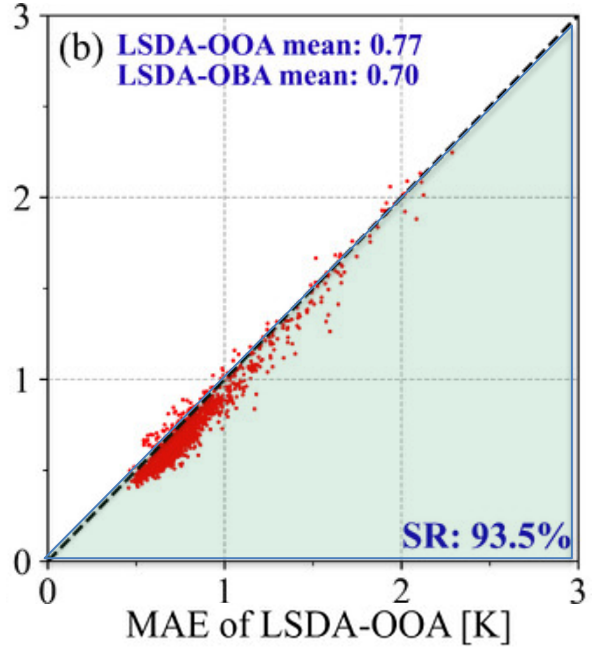
$$\mathbf{B}_z = \left[E\left(\mathbf{x}_a\right) - E\left(\mathbf{x}_b\right)\right]\left[E\left(\mathbf{x}_a\right) - E\left(\mathbf{x}_b\right)\right]^{\mathrm{T}}$$

$$\mathbf{R}_z = \left[\mathrm{O2L}\left(H\left(\mathbf{x}_a\right) + \boldsymbol{\epsilon}\right) - E\left(\mathbf{x}_a\right)\right]\left[\mathrm{O2L}\left(H\left(\mathbf{x}_a\right) + \boldsymbol{\epsilon}\right) - E\left(\mathbf{x}_a\right)\right]^{\mathrm{T}}$$

sparse observations



dense observations



Practical example: LS3D-Var using a global mutlivariate multilevel atmospheric DA model using a CNN AE and a UNet forecasting model

Single observation experiments:

- Ensemble approach: 100 members:
  - Sampling perturbed background members:

$$z'_{bi} = z_{bi} + \epsilon_{bi}, \qquad \epsilon_{bi} \sim \mathcal{N}(0, \sigma_{bi})$$

σ_{bi} … climatological standard deviation of a latent vector element
(i.e. square root of corresponding diagonal element in $\mathbf{B}_z$)

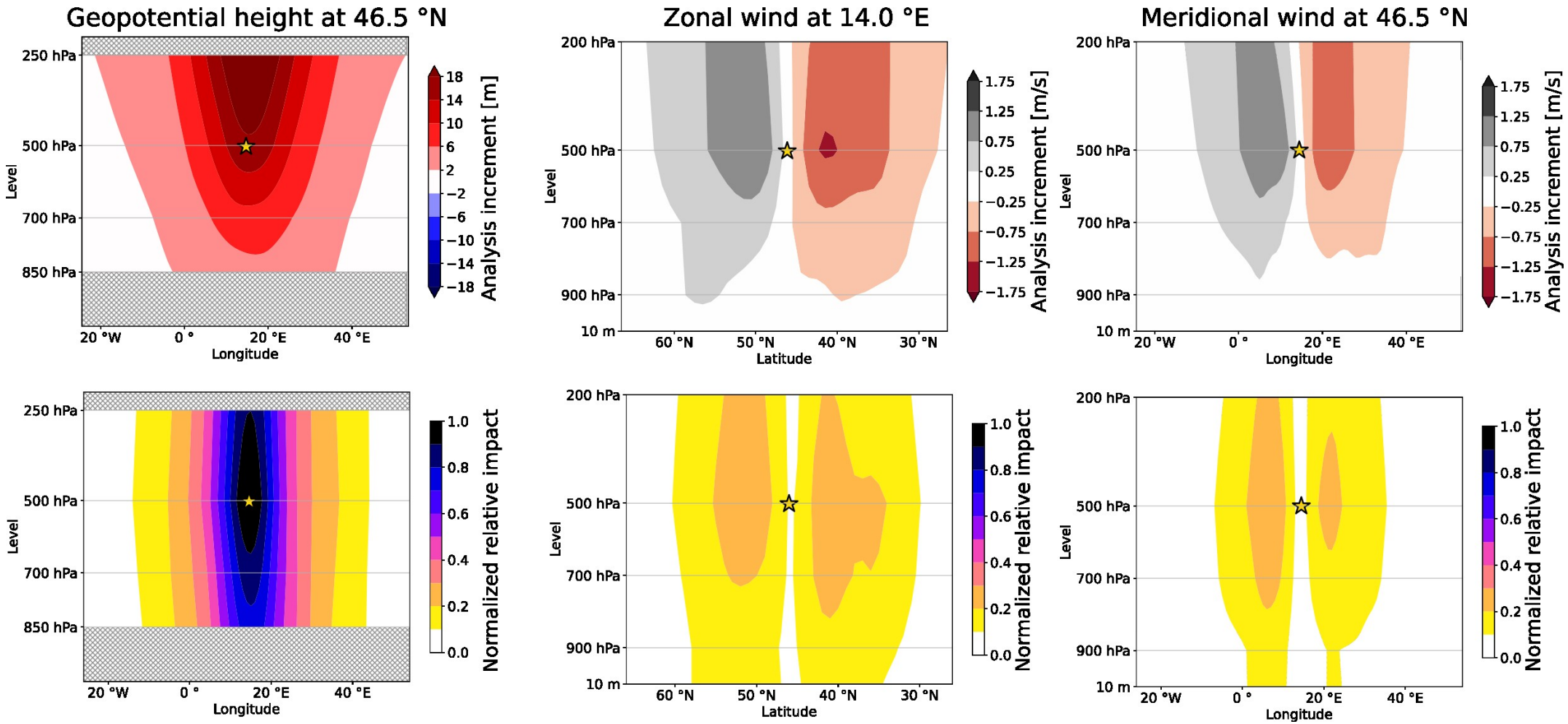
  - Sampling perturbed observations:

$$\mathbf{y}' = H\left(\overline{D(\mathbf{z}'_b)}\right) + \mathbf{d} + \boldsymbol{\epsilon}_o, \qquad \boldsymbol{\epsilon}_o \sim \mathcal{N}(\mathbf{0}, \sigma_o)$$

$\mathbf{d}$ … predefined observation departure
σ_o … predefined observation standard deviation

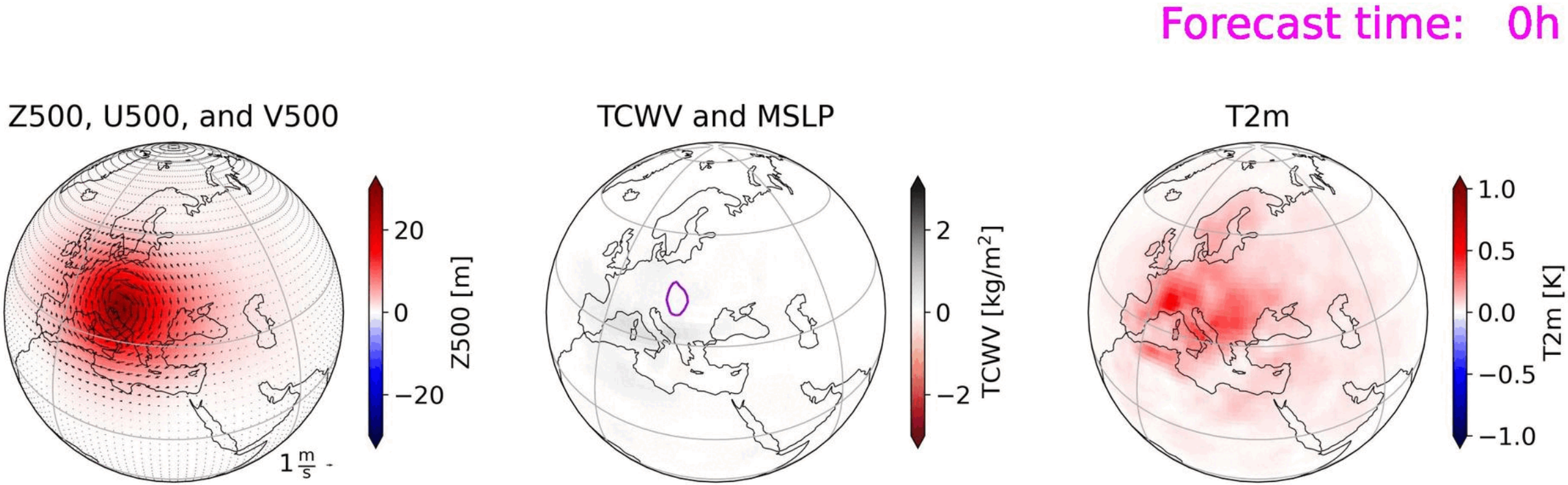
  - We will study analysis increments: $\delta\mathbf{x}_a = \overline{D(\mathbf{z}_a)} - \overline{D(\mathbf{z}'_b)}$

- Realistic vertical increment propagation with max. observation impact at 500 hpa
  - obs. impact = ana. inc. / ana. std., normalized by its value for Z500 at obs. loc



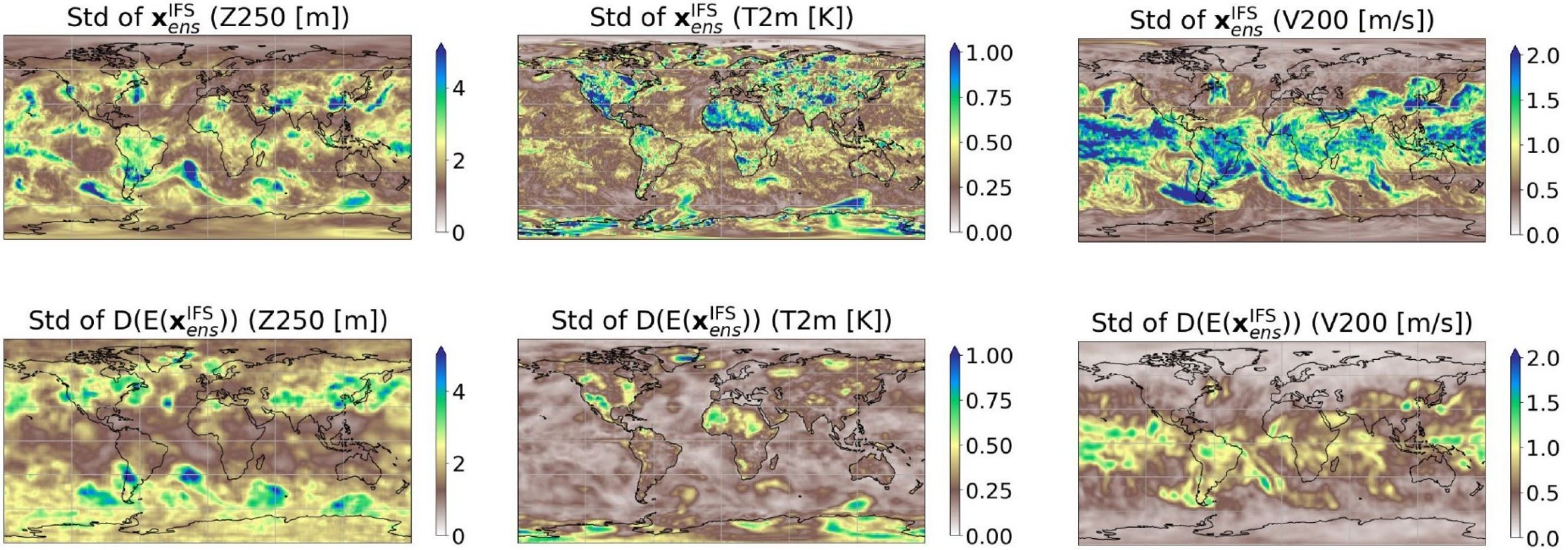
Practical example – evolution

- Difference between respective forecasts initialised by the analysis and the background
  - Eastward propagation of positive Z500 increment
  - Emerging cold front



Challenges of using LSDA operationally – representing EDA variances

- Climatology-based **B**-matrices tend to overestimate variances
- IFS uses hybrid EDA with combined ensemble-derived and climatology-derived variances
- AE is trained on climatological data,
climatological variability ≫ EDA ensemble of backgrounds variability
  - Can AE resolve EDA variances?



Challenges of using LSDA operationally – representing EDA variances

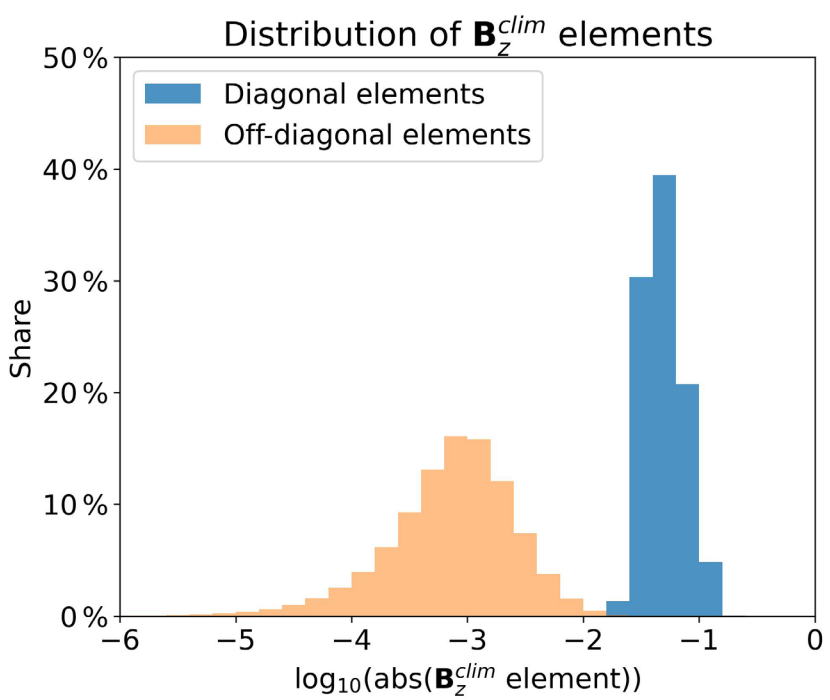
IFS EDA-based $\mathbf{B}_z$:

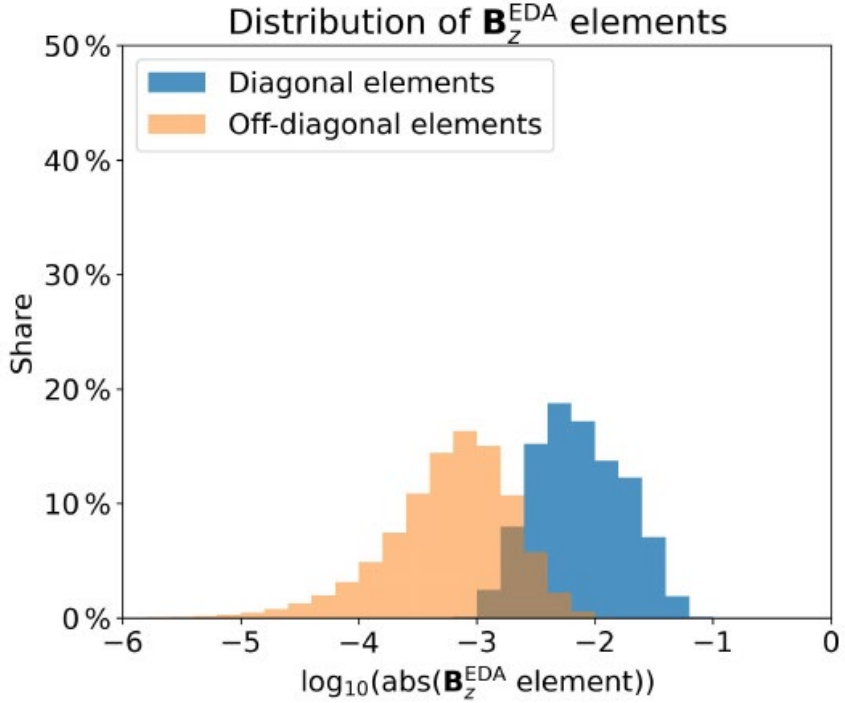
$$\mathbf{B}_z^{\mathrm{EDA}} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\mathbf{z}_{b,i}^{\mathrm{IFS}} - \left\langle \mathbf{z}_b^{\mathrm{IFS}} \right\rangle\right)\left(\mathbf{z}_{b,i}^{\mathrm{IFS}} - \left\langle \mathbf{z}_b^{\mathrm{IFS}} \right\rangle\right)^{\top},$$

where $N = 50$ and $\mathbf{z}_{b,i}^{\mathrm{IFS}} = E\left(\mathbf{x}_{b,i}^{\mathrm{IFS}}\right)$

$\mathbf{B}_z$ still quasi-diagonal:

- diagonals smaller than for climatological $\mathbf{B}_z$
- off-diagonals comparable to climatological $\mathbf{B}_z$
  → we assume they represent sampling noise, keep using only diagonals for inverse computation

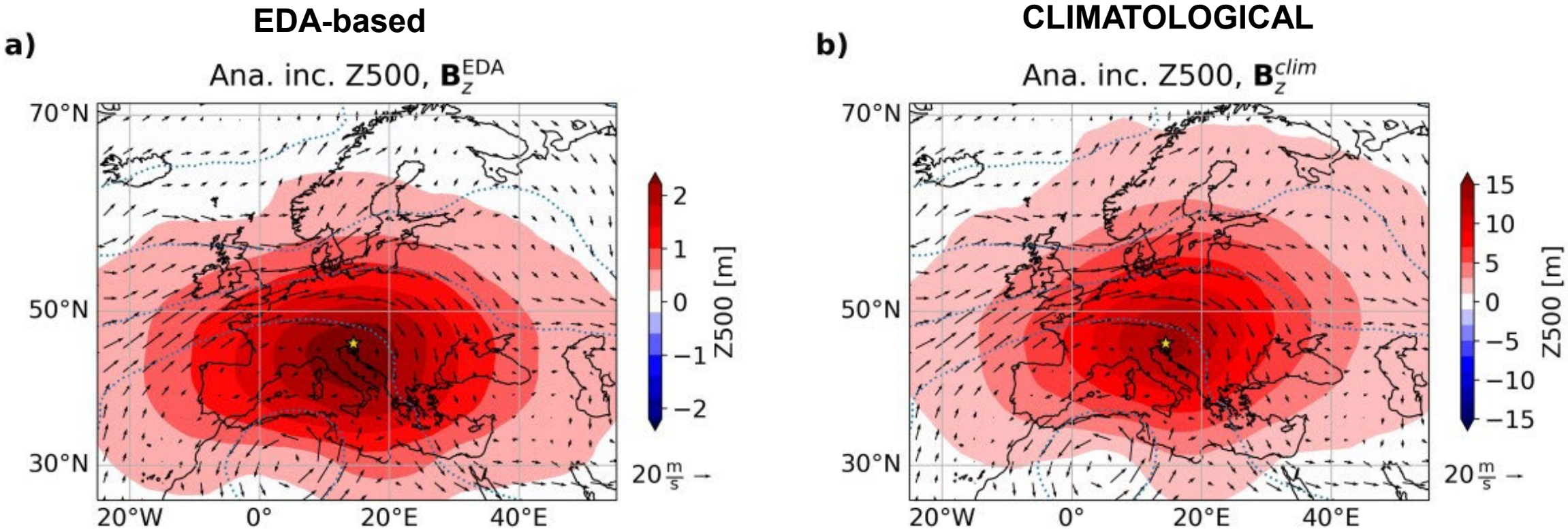


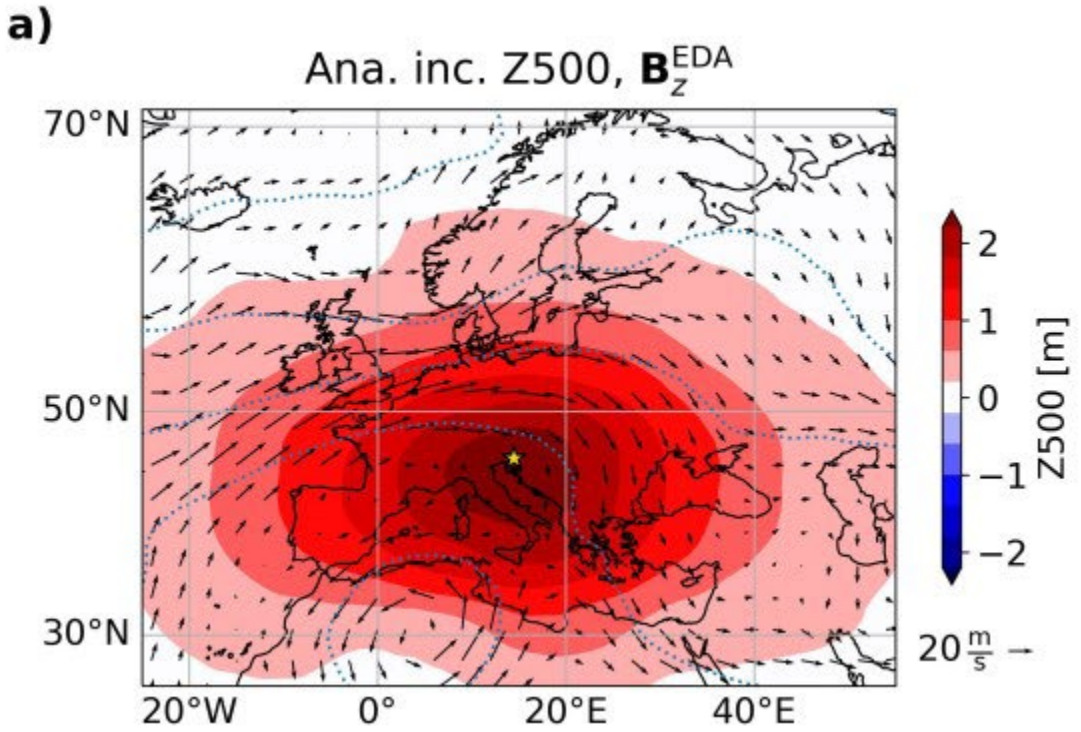


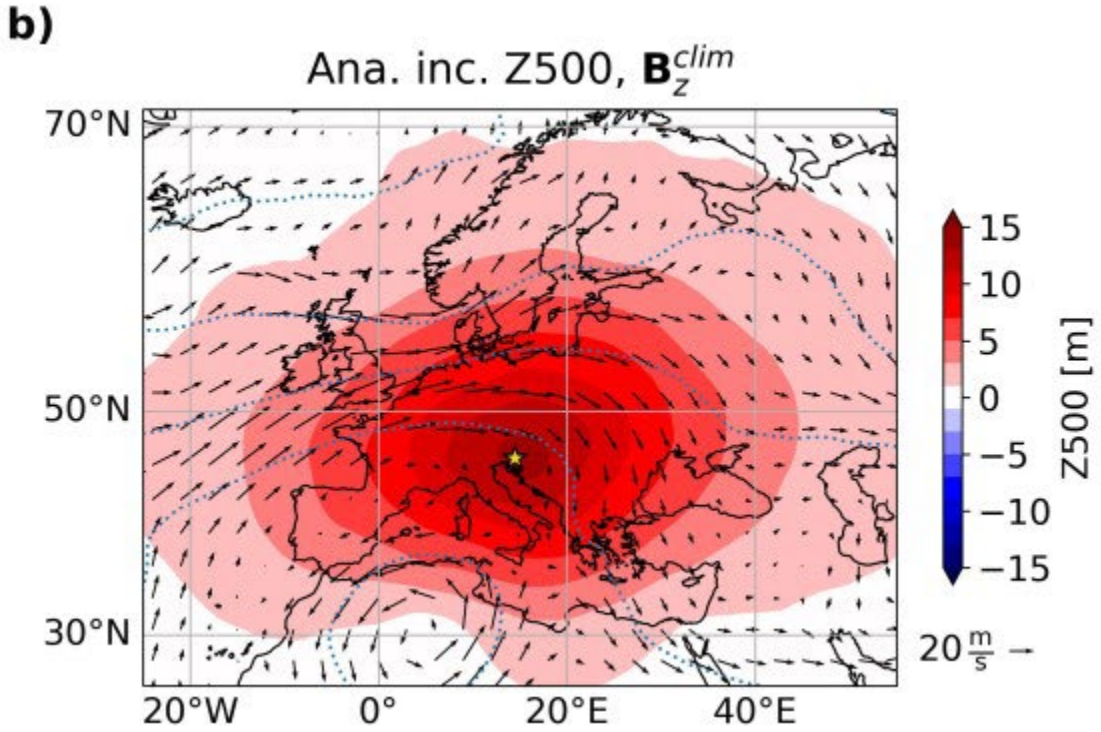
ECMWF

Challenges of using LSDA operationally – representing EDA variances

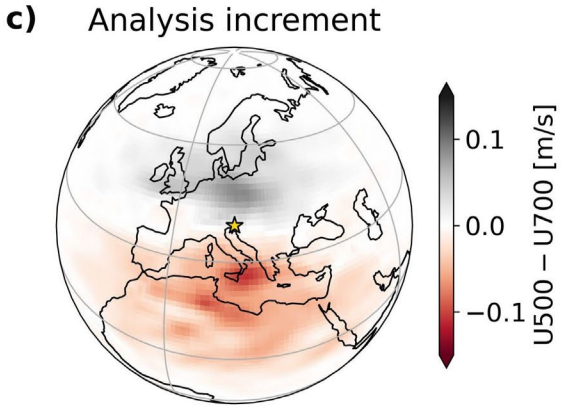
Using climatological **B** and EDA **B**
yields similar analysis increments!

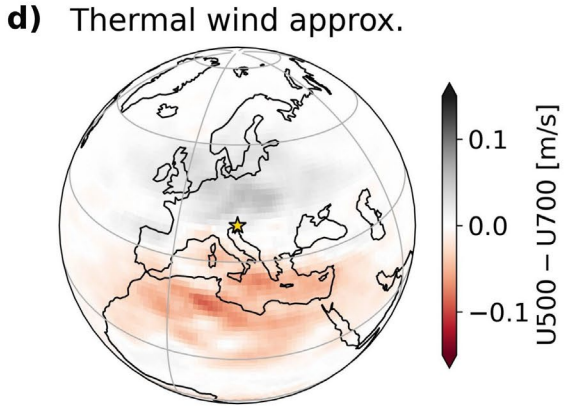


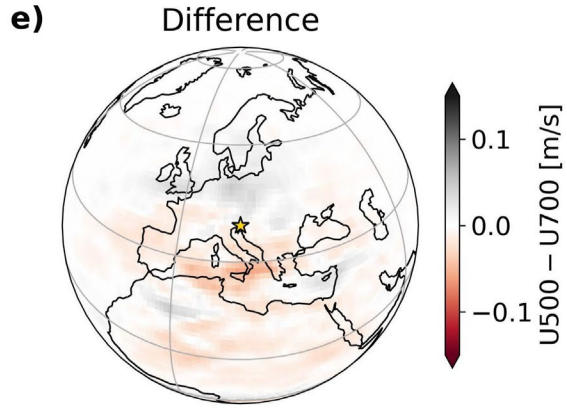




- Flow dependence barely affected
- All in all encouraging results given poorly resolved variances







Motivation

2) Improve flow-dependency:

In operational DA, the flow-dependency of analysis inc. is obtained through:

b) Control variable transform (CVT). Nonlinear equations, linearised around the background flow $\boldsymbol{x}_b$ are part of the balance transform $\mathbf{K}$:

- Linearised nonlinear balance equation

$$\nabla\left(\left(\mathbf{v}_\psi\cdot\nabla\right)\mathbf{v}_\psi+f\mathbf{k}\times\nabla\psi\right)=-\nabla^2\Phi \longrightarrow \nabla\left(\left(\delta\mathbf{v}_\psi\cdot\nabla\right)\mathbf{v}_{b\psi}+\left(\mathbf{v}_{b\psi}\cdot\nabla\right)\delta\mathbf{v}_\psi+f\mathbf{k}\times\nabla\delta\psi\right)=-\nabla^2\delta\Phi$$

- Linearised relative humidity equation (via Clausius-Clapeyron eq.)

$$\mathrm{RH}=\frac{e}{e_\mathrm{s}(T)}=\frac{R_\mathrm{d}}{R_\mathrm{v}}\frac{p}{e_{\mathrm{s}0}\exp\left[\frac{L_\mathrm{c}}{R_\mathrm{v}}\left(\frac{1}{T_0}-\frac{1}{T}\right)\right]}q \longrightarrow \delta\mathrm{RH}=-\frac{L_\mathrm{c}}{R_\mathrm{v}}\frac{\mathrm{RH_g}}{T_\mathrm{g}^2}\delta T+\frac{1}{q_\mathrm{s}(T_\mathrm{g})}\delta q$$

- Linearised quasi-geostrophic omega equation

$$\left[\sigma\nabla^2\omega+f_0^2\frac{\partial^2}{\partial p^2}\right]\delta\omega=-2\cdot\nabla\delta\mathbf{Q}\text{, where}$$

$$\delta\mathbf{Q}=(\delta Q_x,\delta Q_y)=-\frac{R}{p}\left(\frac{\partial\delta\mathbf{v}_\psi}{R_e\cos\phi\,\partial\lambda}\cdot\nabla T_b+\frac{\partial\mathbf{v}_{b\psi}}{R_e\cos\phi\,\partial\lambda}\cdot\nabla\delta T,\frac{\partial\delta\mathbf{v}_\psi}{R_e\,\partial\phi}\cdot\nabla T_b+\frac{\partial\mathbf{v}_{b\psi}}{R_e\,\partial\phi}\cdot\nabla\delta T\right)$$

Are 3D-Var and LS3D-Var analyses equivalent?

Assumptions:

- *D* is an *error-free* decoder mapping from latent space to physical space
- *D* is *affine (= linear in all dimensions) over the region traversed during assimilation* $(D = \mathbf{A}z + b)$

Proof of sufficiency:

Let $\mathbf{x}_a$ and $\mathbf{z}_a$ denote the optimal analyses from 3D-Var and LS3D-Var:

$$\nabla_{\mathbf{x}}\mathcal{J}(\mathbf{x}=\mathbf{x}_a) = \mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b) + \mathbf{H}^{\top}\mathbf{R}^{-1}[H(\mathbf{x}_a) - \mathbf{y}] = \mathbf{0}, \quad (1)$$

$$\nabla_{\mathbf{z}}\mathcal{J}_z(\mathbf{z}=\mathbf{z}_a) = \mathbf{B}_z^{-1}(\mathbf{z}_a - \mathbf{z}_b) + \mathbf{J}_D^{\top}\mathbf{H}^{\top}\mathbf{R}^{-1}[H(D(\mathbf{z}_a)) - \mathbf{y}] = \mathbf{0}, \quad (2)$$

where $\mathbf{J}_D = (\partial D/\partial \boldsymbol{z})$ i.e. a linearised *D* at $\mathbf{z}_a$, and $\mathbf{H} = (\partial H/\partial \boldsymbol{x})$ is the linearised *H* at $\mathbf{x}_a$

Multiply (1) by **B** and (2) by $\mathbf{J}_D\mathbf{B}_z$ from the left

$$(\mathbf{x}_a - \mathbf{x}_b) + \mathbf{H}^{\top}\mathbf{R}^{-1}[H(\mathbf{x}_a) - \mathbf{y}] = \mathbf{0}, \quad (3)$$

$$\mathbf{J}_D(\mathbf{z}_a - \mathbf{z}_b) + \mathbf{J}_D\mathbf{B}_z\mathbf{J}_D^{\top}\mathbf{H}^{\top}\mathbf{R}^{-1}[H(D(\mathbf{z}_a)) - \mathbf{y}] = \mathbf{0} \quad (4)$$

Adapted from Fan et al. (2026a)

$$(\mathbf{x}_a - \mathbf{x}_b) + \mathbf{H}^\top \mathbf{R}^{-1} \left[H(\mathbf{x}_a) - \mathbf{y} \right] = \mathbf{0}, \tag{3}$$

$$\mathbf{J}_D (\mathbf{z}_a - \mathbf{z}_b) + \mathbf{J}_D \mathbf{B}_z \mathbf{J}_D^\top \mathbf{H}^\top \mathbf{R}^{-1} \left[H(D(\mathbf{z}_a)) - \mathbf{y} \right] = \mathbf{0} \tag{4}$$

D is affine throughout the assimilation process ($\mathbf{J}_D(\mathbf{z}_a) = \mathbf{J}_D(\mathbf{z}_b) = \mathbf{J}_D(\mathbf{z}_t)$):

$$\begin{aligned}
\mathbf{J}_D \mathbf{B}_z \mathbf{J}_D^\top &= \mathbf{J}_D \mathbb{E}\left[(\mathbf{z}_t - \mathbf{z}_b)(\mathbf{z}_t - \mathbf{z}_b)^\top \right] \mathbf{J}_D^\top \\
&= \mathbb{E}\left[(\mathbf{J}_D (\mathbf{z}_t - \mathbf{z}_b))((\mathbf{z}_t - \mathbf{z}_b))^\top \mathbf{J}_D^\top \right] \\
&= \mathbb{E}\left[(D(\mathbf{z}_t) - D(\mathbf{z}_b))(D(\mathbf{z}_t) - D(\mathbf{z}_b))^\top \right] \\
&= \mathbb{E}\left[(\mathbf{x}_t - \mathbf{x}_b)(\mathbf{x}_t - \mathbf{x}_b)^\top \right] \\
&= \mathbf{B}
\end{aligned} \tag{5}$$

$$\begin{aligned}
\mathbf{x}_a^{\mathrm{LS3DVar}} - \mathbf{x}_b &= D(\mathbf{z}_a) - \{ D(\mathbf{z}_a) + \mathbf{J}_D [\mathbf{z}_b - \mathbf{z}_a] \} \\
&= \mathbf{J}_D [\mathbf{z}_a - \mathbf{z}_b]
\end{aligned} \tag{6}$$

Insert (5,6) into (4) and multiply by $\mathbf{B}^{-1}$

$$\mathbf{B}^{-1} \left(\mathbf{x}_a^{\mathrm{LS3DVar}} - \mathbf{x}_b \right) + \mathbf{H}^\top \mathbf{R}^{-1} \left[H(\mathbf{x}_a) - \mathbf{y} \right] = \mathbf{0} \quad \rightarrow \text{equivalent to (1)}$$

Proof of necessity:

First-order Taylor expansion of D at $\mathbf{x}_a$:

$$\begin{aligned}\mathbf{x}_a - \mathbf{x} &= D(\mathbf{z}_a) - \{D(\mathbf{z}_a) + \mathbf{J}_D\left[\mathbf{z} - \mathbf{z}_a\right] + \boldsymbol{\varepsilon}\} \\ &= \mathbf{J}_D\left[\mathbf{z}_a - \mathbf{z}\right] + \boldsymbol{\varepsilon},\end{aligned}$$

where ε denotes the higher-order residual which depends both on $\boldsymbol{z}_a$ and $\boldsymbol{z}$. Similarly

$$\mathbf{x}_a - \mathbf{x}_t = \mathbf{J}_D\left[\mathbf{z}_a - \mathbf{z}_t\right] + \boldsymbol{\varepsilon}_1, \tag{7}$$

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{J}_D\left[\mathbf{z}_a - \mathbf{z}_b\right] + \boldsymbol{\varepsilon}_2. \tag{8}$$

Long algebraic derivation using (7), (8), and

$$\mathbf{B} = \mathbb{E}\left[(\mathbf{x}_t - \mathbf{x}_b)(\mathbf{x}_t - \mathbf{x}_b)^\top\right]$$

leads to

$$\boldsymbol{\varepsilon}_2 = \mathbf{A}\mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b), \tag{9}$$

where

$$\begin{aligned}\mathbf{A} = &\mathbb{E}\left[(\boldsymbol{\varepsilon}_2 - \boldsymbol{\varepsilon}_1)(\boldsymbol{\varepsilon}_2 - \boldsymbol{\varepsilon}_1)^\top\right] \\ &- \mathbb{E}\left[\left(\mathbf{J}_D\left(\mathbf{z}_t - \mathbf{z}_a\right)\left(\boldsymbol{\varepsilon}_2 - \boldsymbol{\varepsilon}_1\right) + \left(\boldsymbol{\varepsilon}_2 - \boldsymbol{\varepsilon}_1\right)^\top\left(\mathbf{z}_t - \mathbf{z}_a\right)^\top \mathbf{J}_D^\top\right]\right.\end{aligned}.$$

→ (9) can only hold for every $\mathbf{x}_a$ and $\mathbf{x}_b$ if $\boldsymbol{\varepsilon}_1 = \boldsymbol{\varepsilon}_2 = \boldsymbol{0}$ → D needs to be locally affine!

Stratospheric data assimilation

Operationally, ozone is assimilated univariately, i.e. independently from other atmospheric components

Single-observation experiments ($d^{O3(50hPa)} = +10^{-6}$ kg/kg) with LS3D-Var using climatological $\mathbf{B}_z$ and 18-level multivariate GNN AE indicate coupled ozone and temperature increments

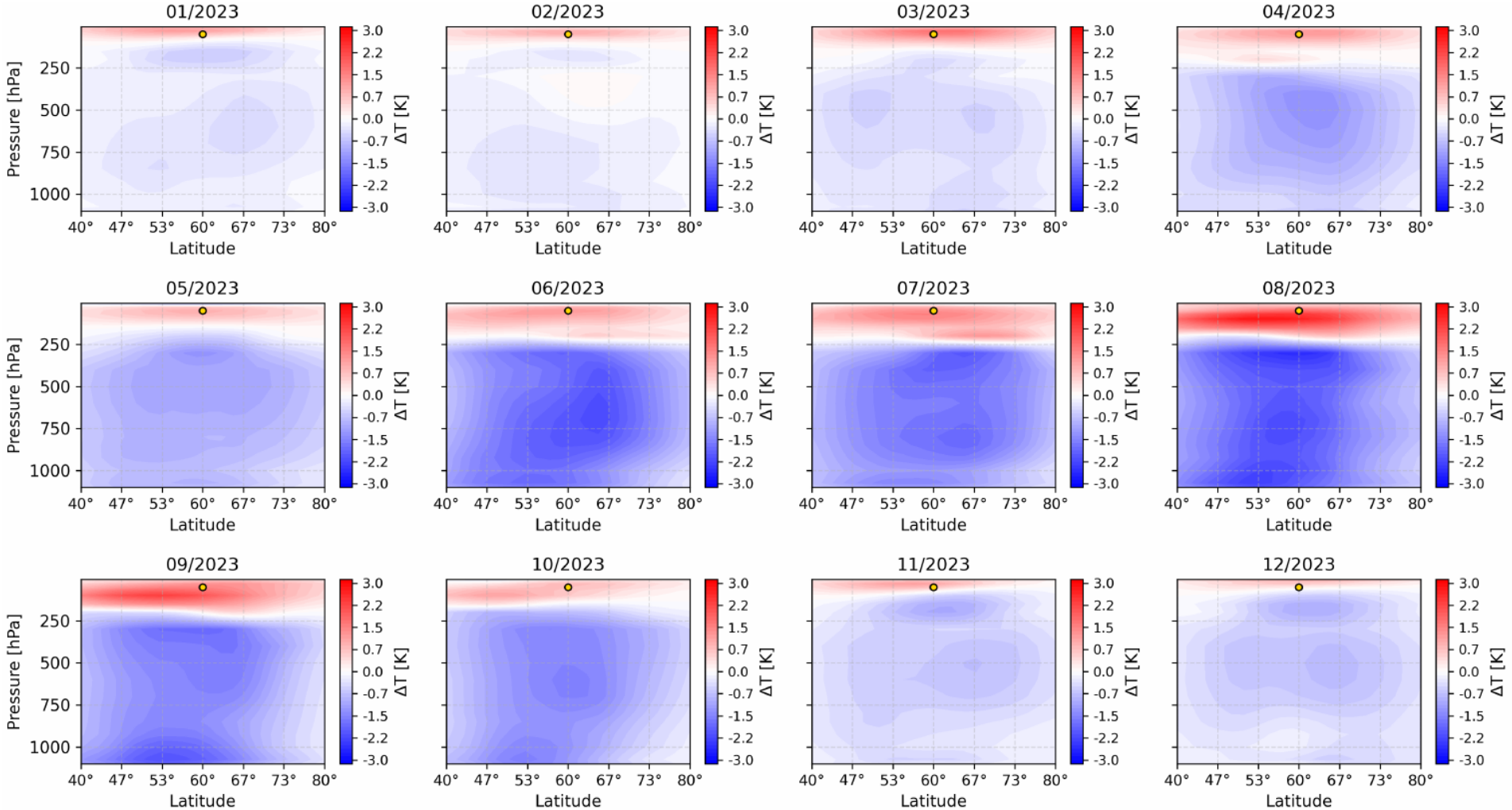


Figure courtesy of Janne Lisa Bouillon Zaplotnik et al. (2026), in prep.

Motivation

3) Improve representation of background-error covariances
→ less imbalanced initial state:

Current operational limitations:

- Equatorial balances not adequately represented ($f \rightarrow 0$)
- Boundary-layer balances missing
- Stratospheric balances including O_3 (mostly) missing
- Background-error covariances do not obey orographical boundaries, land-sea contrasts
- No background-error covariances between different Earth-system components → different DA system for atmosphere, ocean, surface components