# Emulating 3D-Var Data Assimilation using Variational Autoencoder

**Boštjan Melinc**[1], Žiga Zaplotnik[2]

[1]University of Ljubljana, Faculty of Mathematics and Physics
Jadranska 19, 1000 Ljubljana, Slovenia
bostjan.melinc@fmf.uni-lj.si

[2]ECMWF
Shinfield Park, Reading, RG2 9AX Code, United Kingdom
ziga.zaplotnik@ecmwf.int

ISDA Bologna 2023

University of Ljubljana
Faculty of Mathematics and Physics

ECMWF

# Contents

- Motivation
- Design of variational autoencoder (VAE) for $T_{850}$ data
- 3D-Var cost function in the latent space
- Background-error covariance modelling
- Single observation experiments: ($T_{850}$) univariate case
- Single observation experiments: ($Z_{200}$, $u_{200}$, $v_{200}$) multivariate case
- Conclusions and outlook

# Motivation

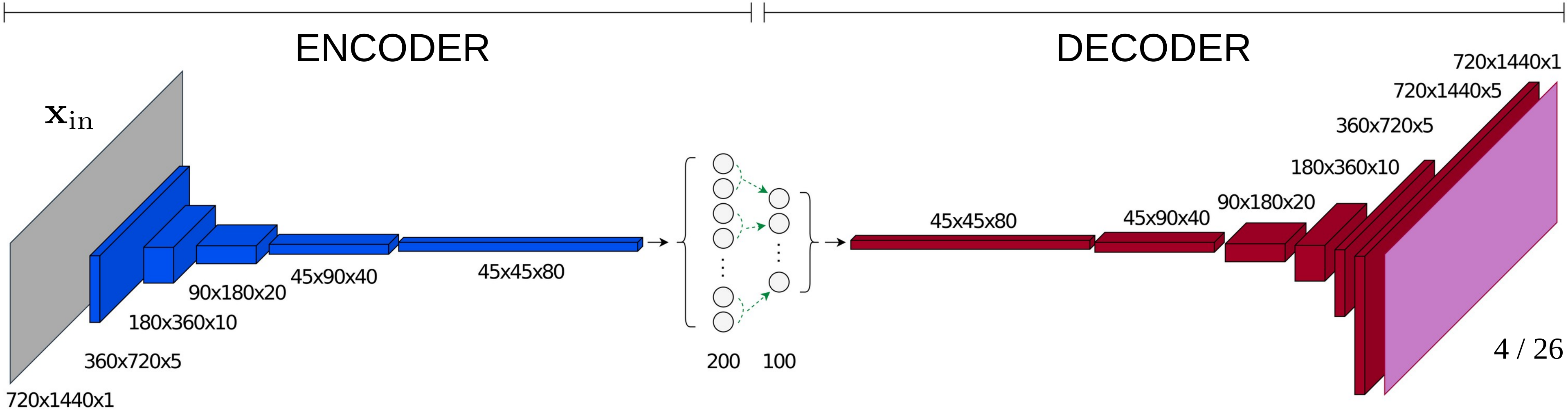
- Variational data assimilation (3D/4D-Var) in current numerical weather prediction (NWP) models is performed in a **control space** defined by analytical transformations utilising manually-defined physical balances
- Weakness: **equatorial balances cannot be adequately represented** using these analytical transformations
- **Idea: Use neural-network-discovered transformations** which describe these balances to perform variational cost function minimisation in a reduced-order latent space
- Recent approaches to neural-network data assimilation (e.g. Mack, 2020; Amendola, 2021; Peyron, 2021) hardly applicable to current NWP, i.e., they require interpolation of (sparse) observations to the (dense) model grid, etc.

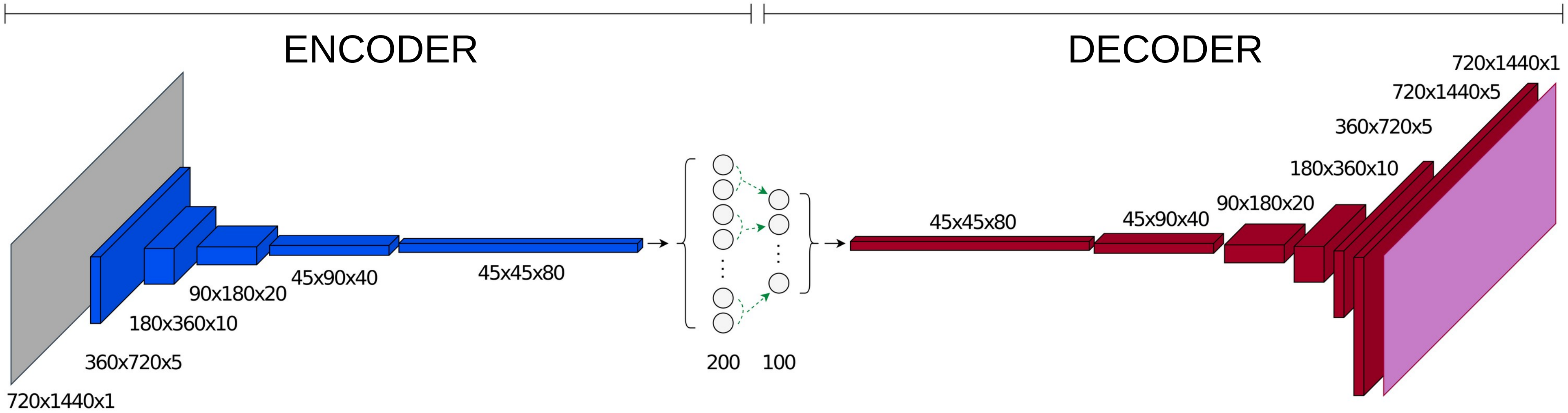
# Variational autoencoder (VAE)

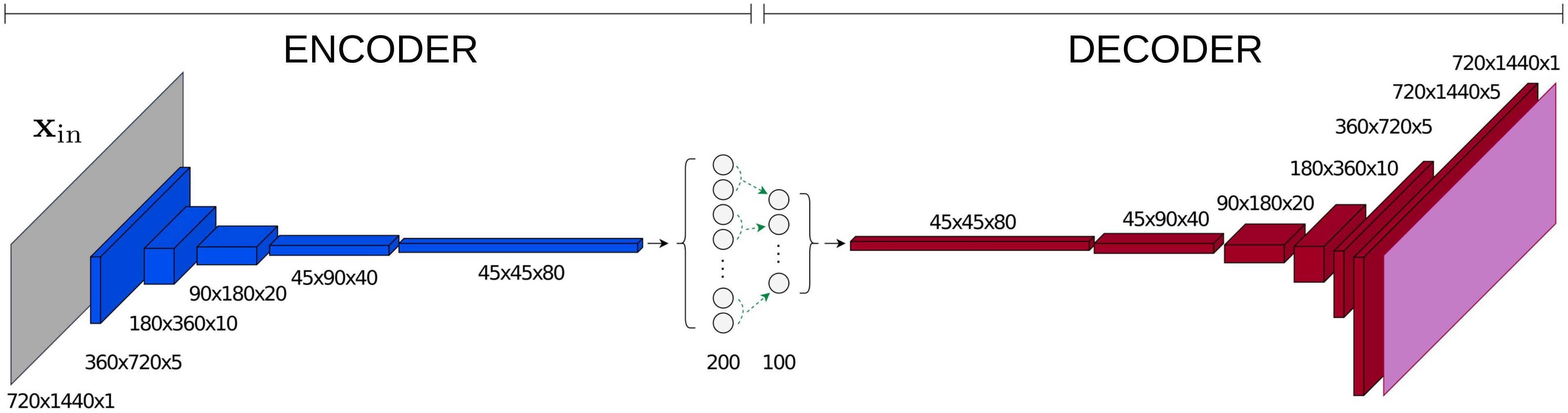
- VAE architecture based on Brohan (2022)
- Input data: daily mean $T_{850}$ from ERA5 reanalysis on latitude-longitude grid (0.25° × 0.25° resolution → 720 × 1440 grid points)
- Data standardisation:

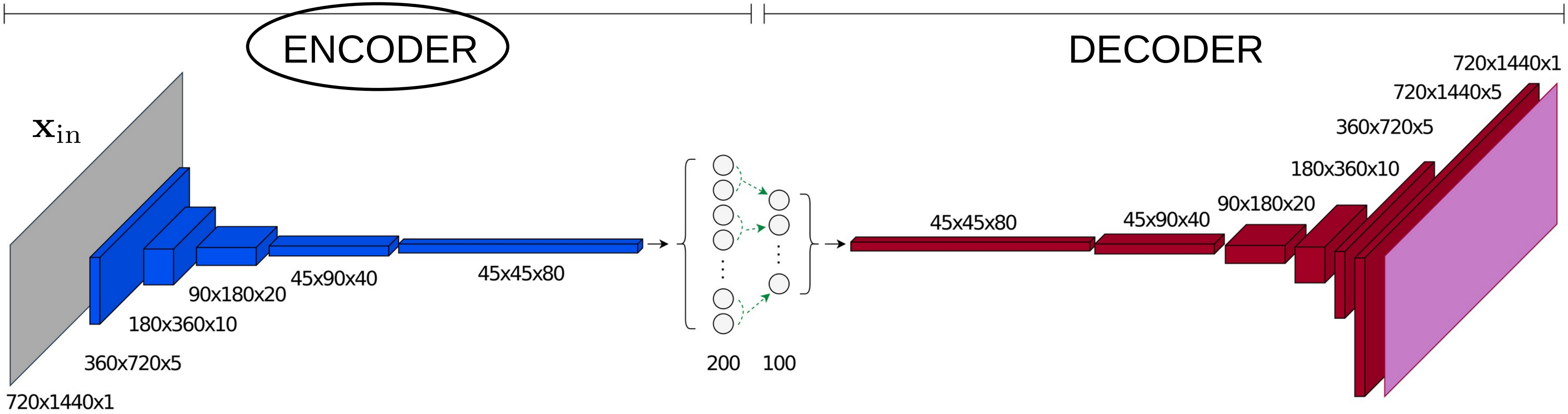
$$\mathbf{x}^{ij} = \frac{T_{850}^{ij} - \texttt{d.c.Mean}(T_{850}^{ij})}{\texttt{d.c.Std}(T_{850}^{ij})}$$

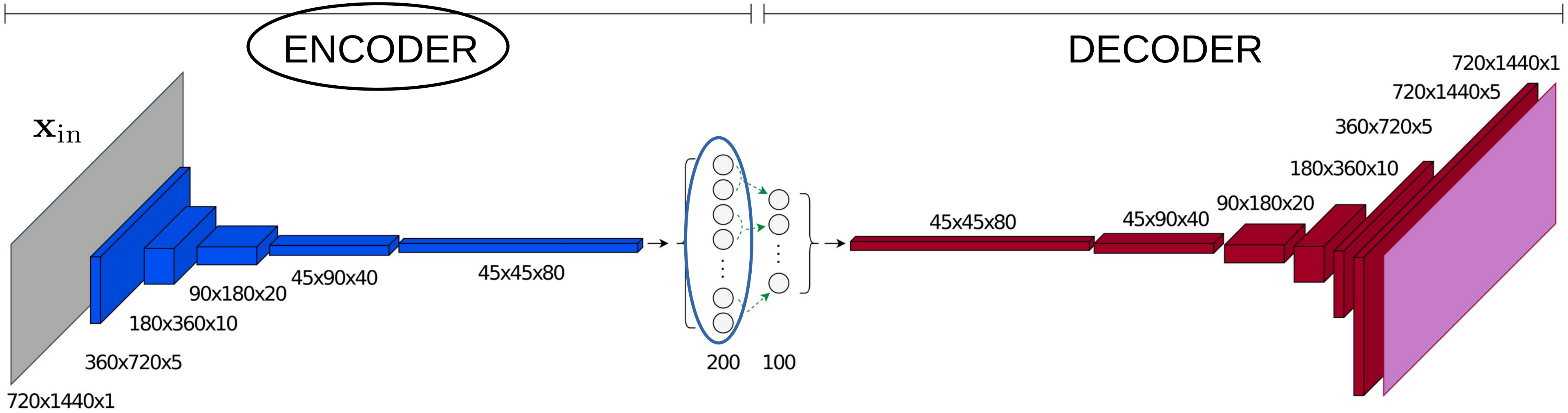
d.c.Mean … climatological mean for day-of-year
d.c.Std … climatological standard deviation for day-of-year

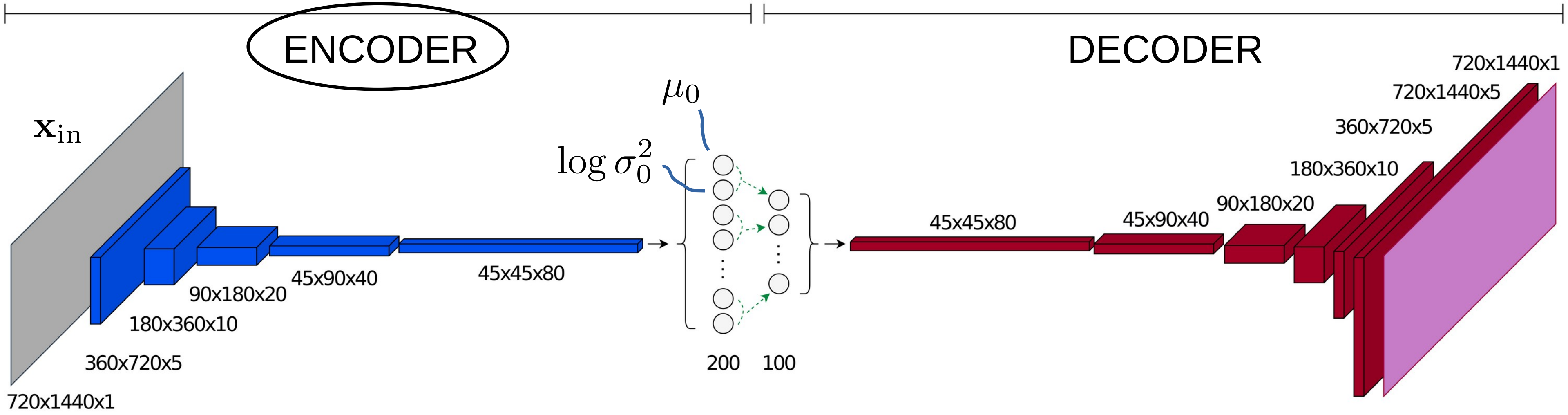


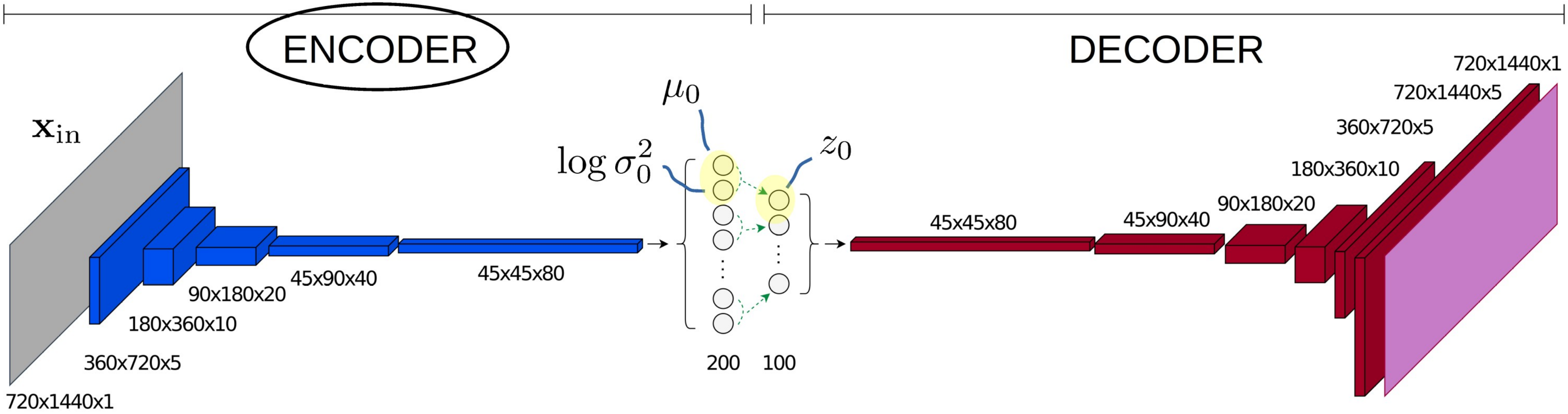

ENCODER
DECODER
720x1440x1
360x720x5
180x360x10
90x180x20
45x90x40
45x45x80
200
100
45x45x80
45x90x40
90x180x20
180x360x10
360x720x5
720x1440x5
720x1440x1


ENCODER
DECODER
$\mathbf{x}_{\mathrm{in}}$
720x1440x1
360x720x5
180x360x10
90x180x20
45x90x40
45x45x80
200
100
45x45x80
45x90x40
90x180x20
180x360x10
360x720x5
720x1440x5
720x1440x1

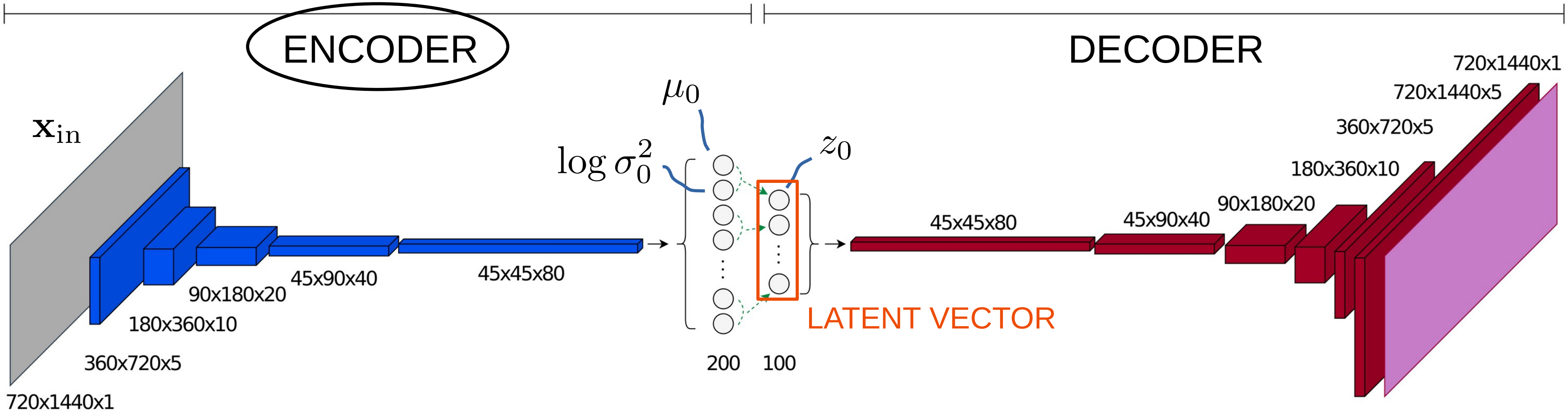

ENCODER
DECODER
$\mathbf{x}_{\text{in}}$
720x1440x1
360x720x5
180x360x10
90x180x20
45x90x40
45x45x80
200
100
45x45x80
45x90x40
90x180x20
180x360x10
360x720x5
720x1440x5
720x1440x1


ENCODER
DECODER
$\mathbf{x}_{\text{in}}$
720x1440x1
360x720x5
180x360x10
90x180x20
45x90x40
45x45x80
200
100
45x45x80
45x90x40
90x180x20
180x360x10
360x720x5
720x1440x5
720x1440x1

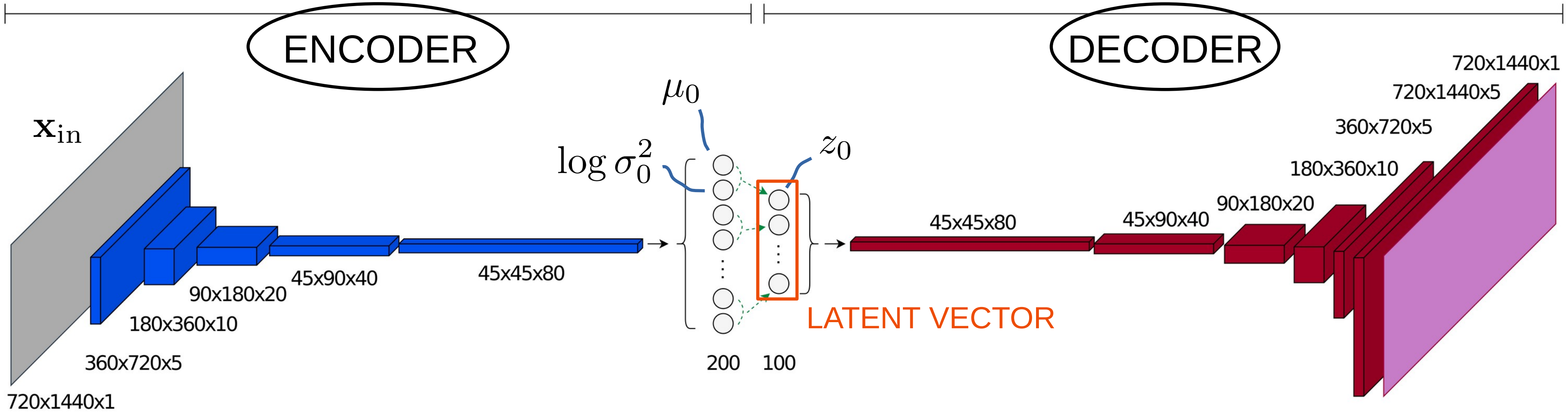

ENCODER
DECODER
$\mathbf{x}_{\text{in}}$
$\mu_0$
$\log \sigma_0^2$
720x1440x1
360x720x5
180x360x10
90x180x20
45x90x40
45x45x80
200
100
45x45x80
45x90x40
90x180x20
180x360x10
360x720x5
720x1440x5
720x1440x1



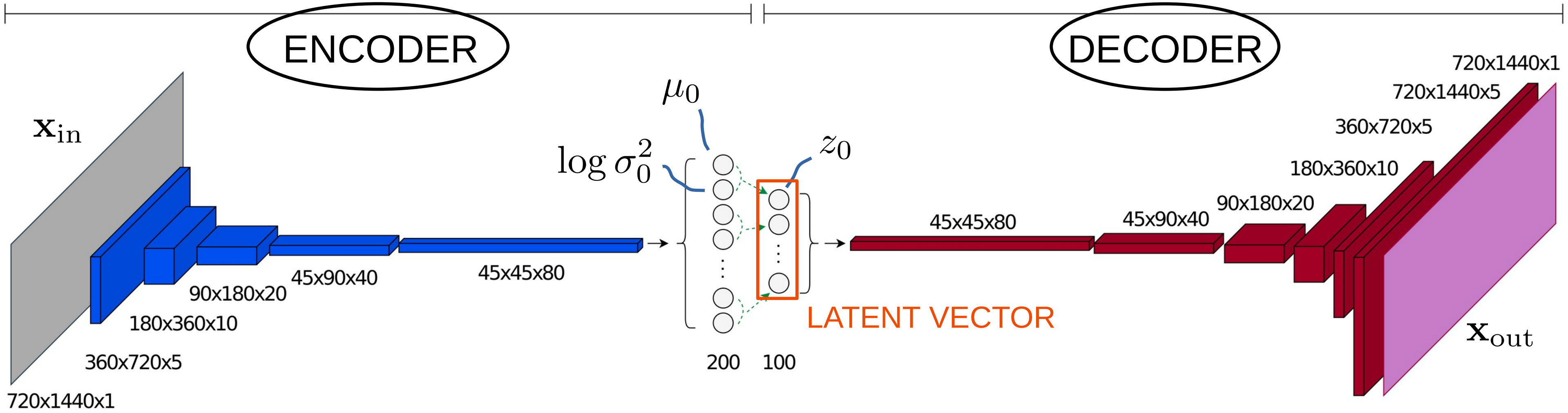

- Sampling $z_i = \mu_i + \hat{z}_i\,\sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0,1)$



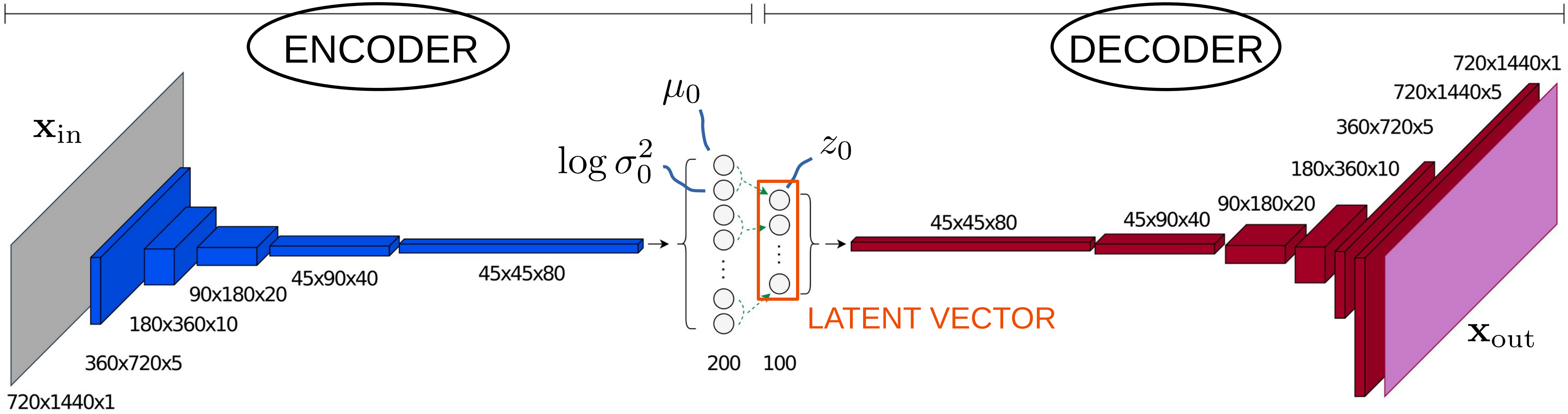

- Sampling $z_i = \mu_i + \hat{z}_i \, \sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0, 1)$
- Latent vector **z** has 100 elements



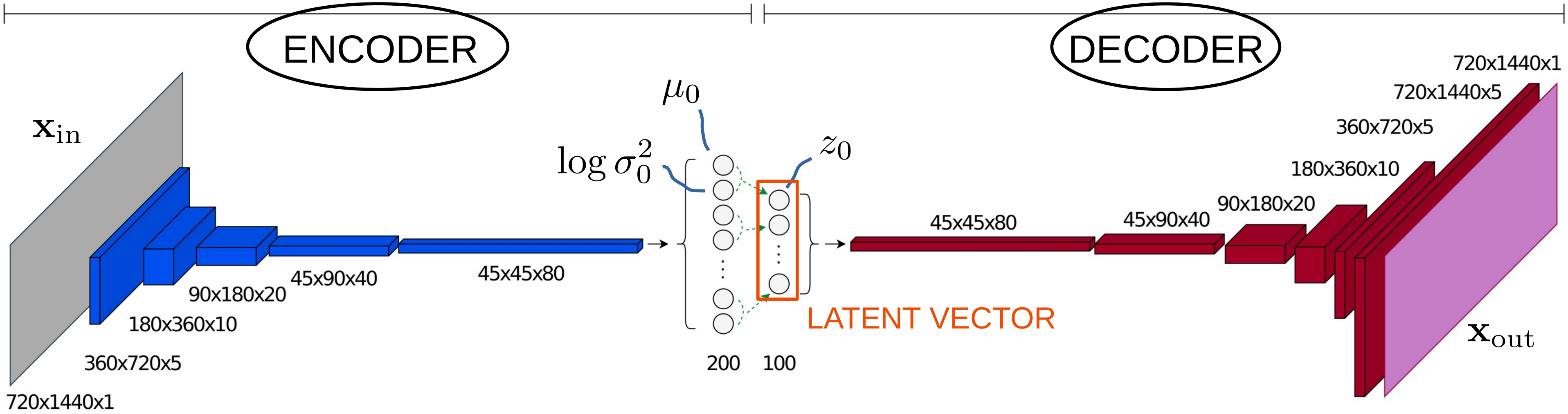

- Sampling $z_i = \mu_i + \hat{z}_i \, \sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0, 1)$
- Latent vector **z** has 100 elements



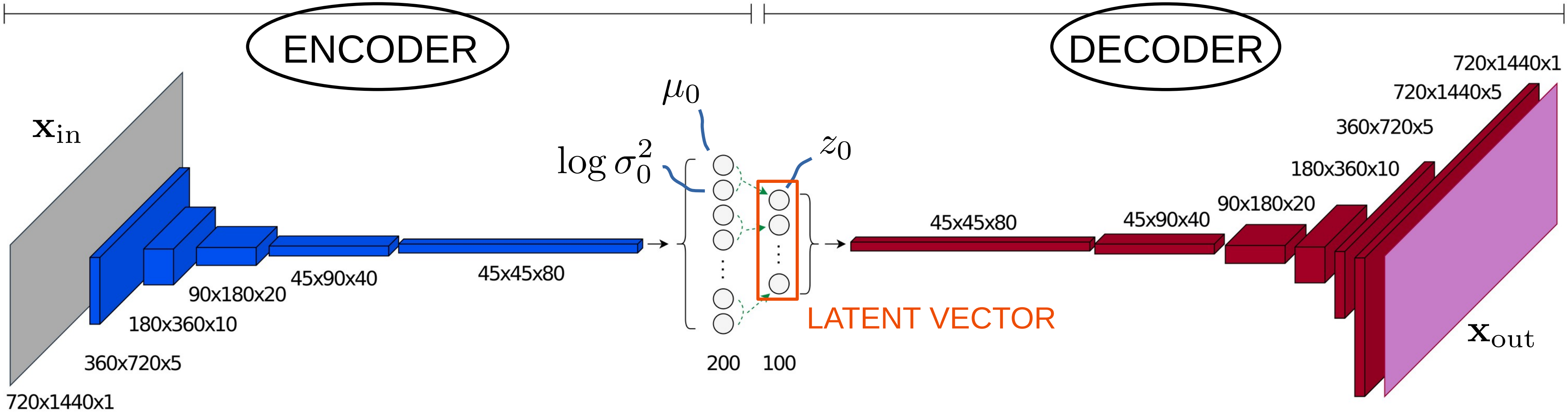

- Sampling $z_i = \mu_i + \hat{z}_i \, \sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0, 1)$
- Latent vector **z** has 100 elements



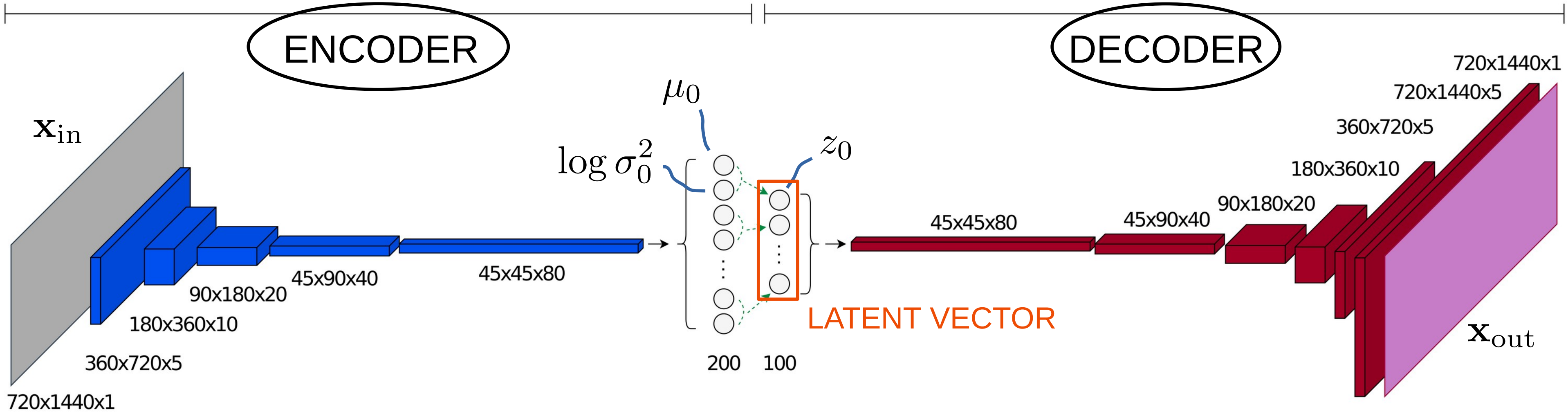

- Sampling $z_i = \mu_i + \hat{z}_i \, \sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0,1)$
- Latent vector **z** has 100 elements
- Training: $\mathcal{L} = \mathcal{L}^{\text{rec}} + \mathcal{L}^{\text{reg}}$
  - Reconstruction loss $\mathcal{L}^{\text{rec}}$
  - Regularisation loss $\mathcal{L}^{\text{reg}}$



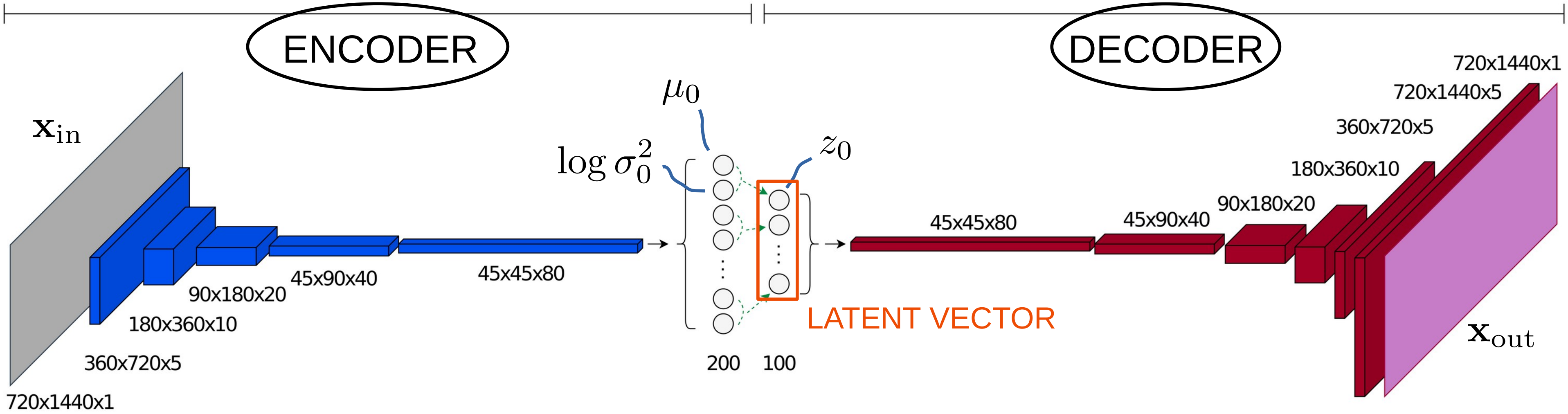

- Sampling $z_i = \mu_i + \hat{z}_i \, \sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0, 1)$
- Latent vector **z** has 100 elements
- Training: $\mathcal{L} = \mathcal{L}^{\text{rec}} + \mathcal{L}^{\text{reg}}$
  - Reconstruction loss $\mathcal{L}^{\text{rec}}$ → $\mathbf{x}_{\text{out}} = D(E(\mathbf{x}_{\text{in}})) \approx \mathbf{x}_{\text{in}}$
  - Regularisation loss $\mathcal{L}^{\text{reg}}$




- Sampling $z_i = \mu_i + \hat{z}_i \, \sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0,1)$
- Latent vector **z** has 100 elements
- Training: $\mathcal{L} = \mathcal{L}^{\text{rec}} + \mathcal{L}^{\text{reg}}$
  - Reconstruction loss $\mathcal{L}^{\text{rec}}$ → $\mathbf{x}_{\text{out}} = D(E(\mathbf{x}_{\text{in}})) \approx \mathbf{x}_{\text{in}}$
  - Regularisation loss $\mathcal{L}^{\text{reg}}$ → $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$

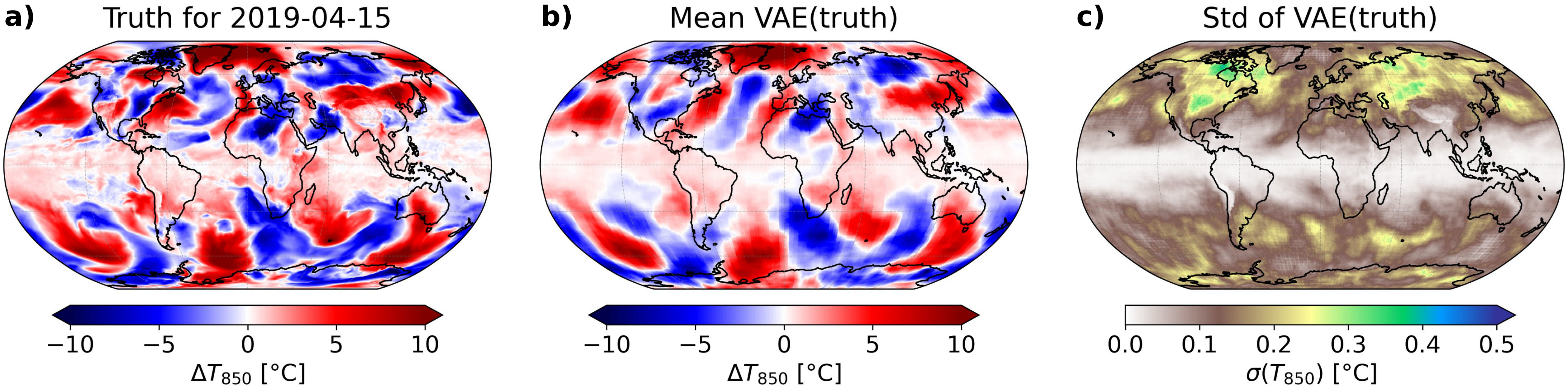



- Sampling $z_i = \mu_i + \hat{z}_i\, \sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0, 1)$
- Latent vector **z** has 100 elements
- Training: $\mathcal{L} = \mathcal{L}^{\text{rec}} + \mathcal{L}^{\text{reg}}$
  - Reconstruction loss $\mathcal{L}^{\text{rec}}$ → $\mathbf{x}_{\text{out}} = D(E(\mathbf{x}_{\text{in}})) \approx \mathbf{x}_{\text{in}}$
  - Regularisation loss $\mathcal{L}^{\text{reg}}$ → $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
- Regularisation ensures **Gaussian properties of the latent space vector** required of variational DA, and its **smoothness**




- Sampling $z_i = \mu_i + \hat{z}_i\,\sigma_i, \quad \hat{z}_i \sim \mathcal{N}(0,1)$
- Latent vector **z** has 100 elements
- Training: $\mathcal{L} = \mathcal{L}^{\text{rec}} + \mathcal{L}^{\text{reg}}$
  - Reconstruction loss $\mathcal{L}^{\text{rec}}$ → $\mathbf{x}_{\text{out}} = D(E(\mathbf{x}_{\text{in}})) \approx \mathbf{x}_{\text{in}}$
  - Regularisation loss $\mathcal{L}^{\text{reg}}$ → $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
- Regularisation ensures **Gaussian properties of the latent space vector** required of variational DA, and its **smoothness**
- Training set 1979-2014, validation set 2015-2018, test set 2019-2022

# Representation of temperature fields with VAE



# 3D-Var cost function

- Assumptions:
  - background and observations are independent
  - their errors are Gaussian
- Cost function:

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

**x** … state vector in the grid point space
$\mathbf{x}_b$ ... background vector
**B** … background-error covariance matrix
**y** ... observation vector
*H* … observation operator
**R** … observation-error covariance matrix

- Cost function:

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

- Cost function in latent space:

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

**z** … latent vector
$\mathbf{z}_b$ **...** background defined in latent space
$\mathbf{B}_z$ … background-error covariance matrix
**y ...** observations vector
*H* … observation operator
*D* … decoder
**R** … observation-error covariance matrix

- Cost function:

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

- Cost function in latent space:

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

**z** … latent vector
$\mathbf{z}_b$ ... background defined in latent space
$\mathbf{B}_z$ … background-error covariance matrix
**y** ... observations vector
*H* … observation operator
*D* … decoder
**R** … observation-error covariance matrix

- Cost function:

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

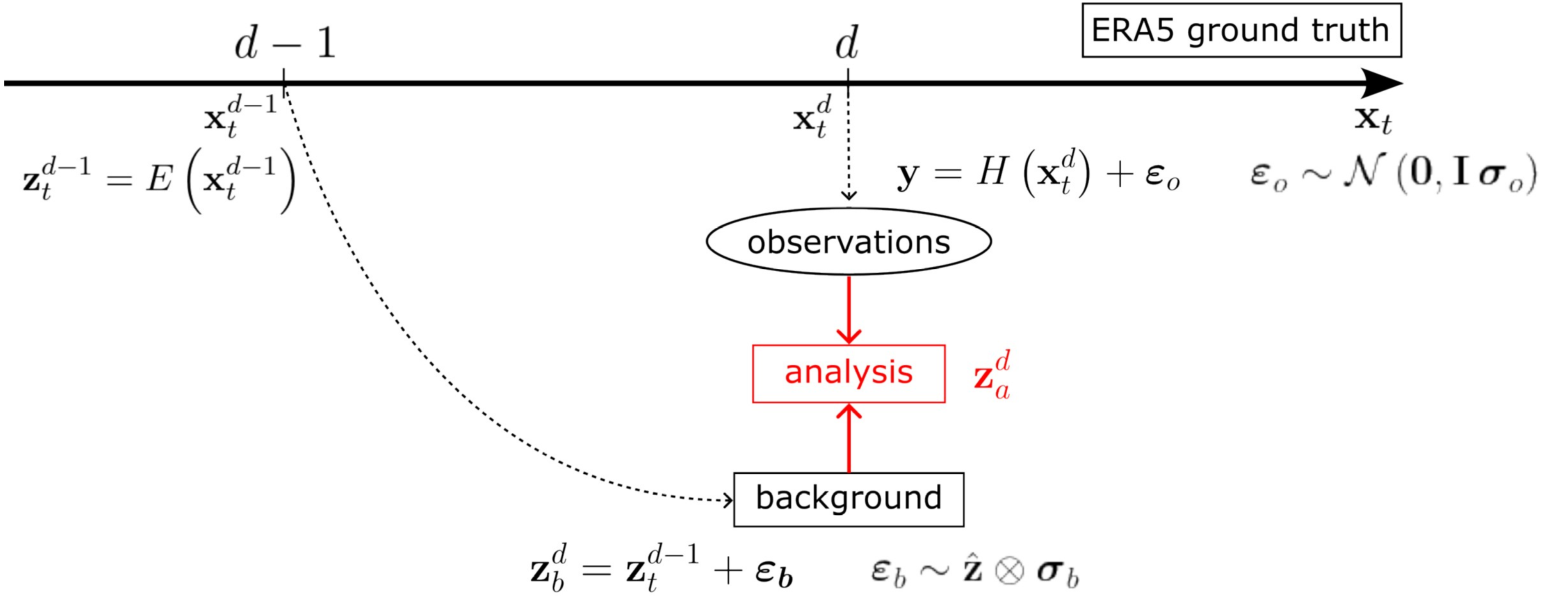
- Cost function in latent space:

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

**z** … latent vector
$\mathbf{z}_b$ **...** background defined in latent space
$\mathbf{B}_z$ … background-error covariance matrix
**y** **...** observations vector
*H* … observation operator
*D* … decoder
**R** … observation-error covariance matrix

# Setup of observing system simulation experiments

- Single assimilation cycle
- **Background simulated from ground truth for previous day** (*d-1*)
- **Observations simulated from ground truth for present day** (*d*)
- Ensemble of data assimilations: 150 ensemble members for background (perturbed according to $\mathbf{B}_z$) and observations (perturbed according to $\mathbf{R}$)

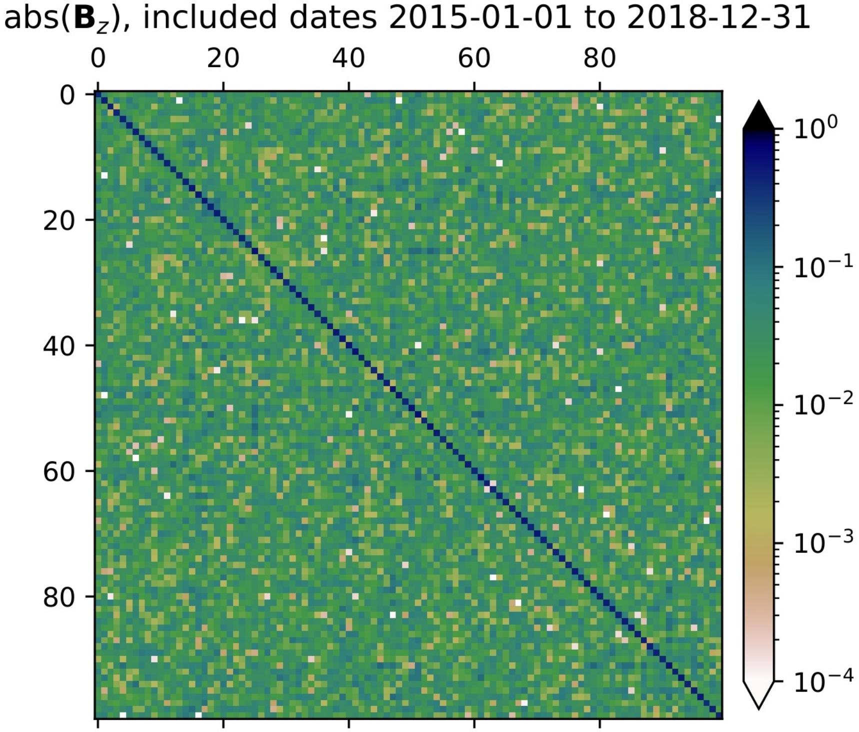


# Background-error covariance matrix

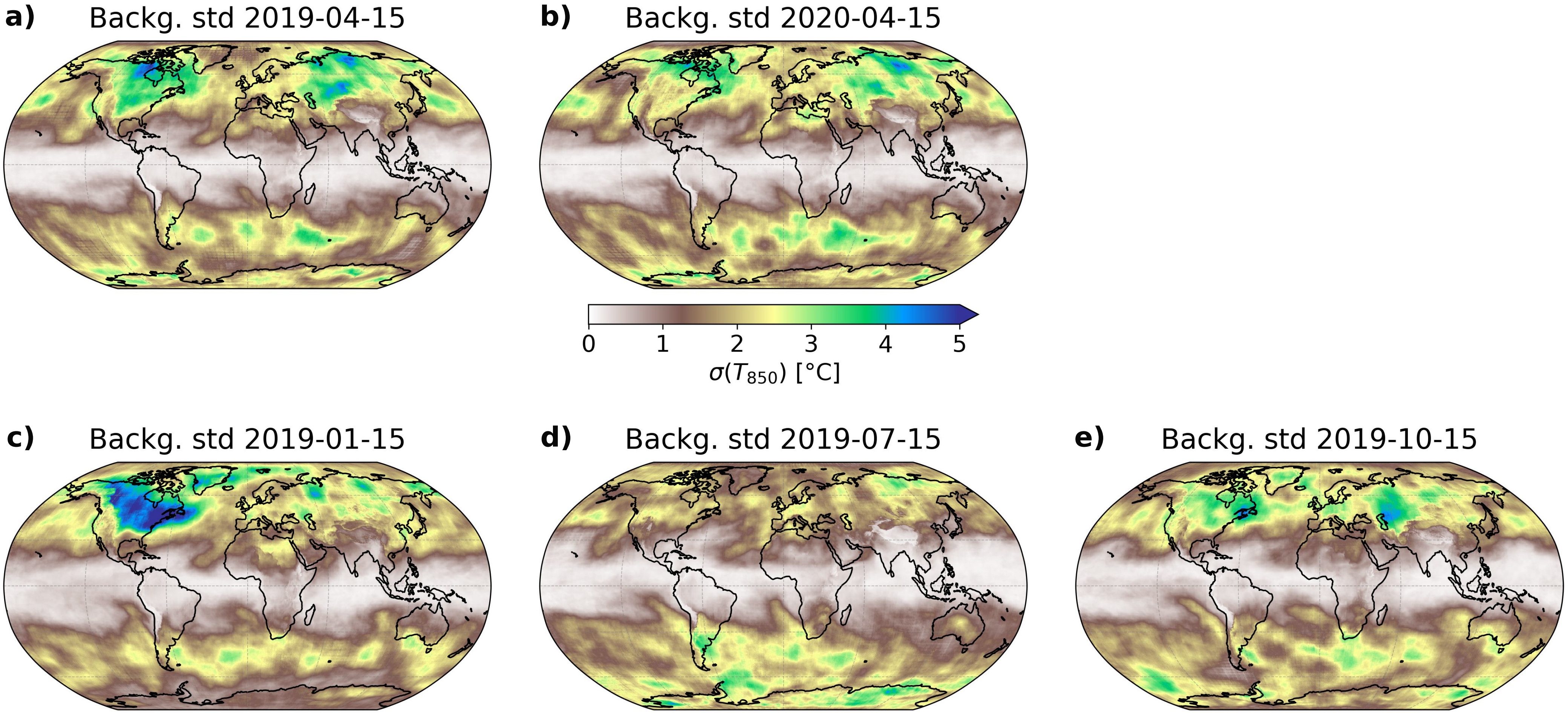
$$\mathbf{B}_z = \left\langle (\mathbf{z}_t - \mathbf{z}_b)\ (\mathbf{z}_t - \mathbf{z}_b)^{\mathrm{T}} \right\rangle$$
$$= \left\langle \left(\mathbf{z}_t^d - \mathbf{z}_t^{d-1}\right)\ \left(\mathbf{z}_t^d - \mathbf{z}_t^{d-1}\right)^T \right\rangle$$

- $\mathbf{B}_z$ quasi-diagonal => we **only use the diagonal elements** for its inverse
- Sampling perturbed background latent vectors:

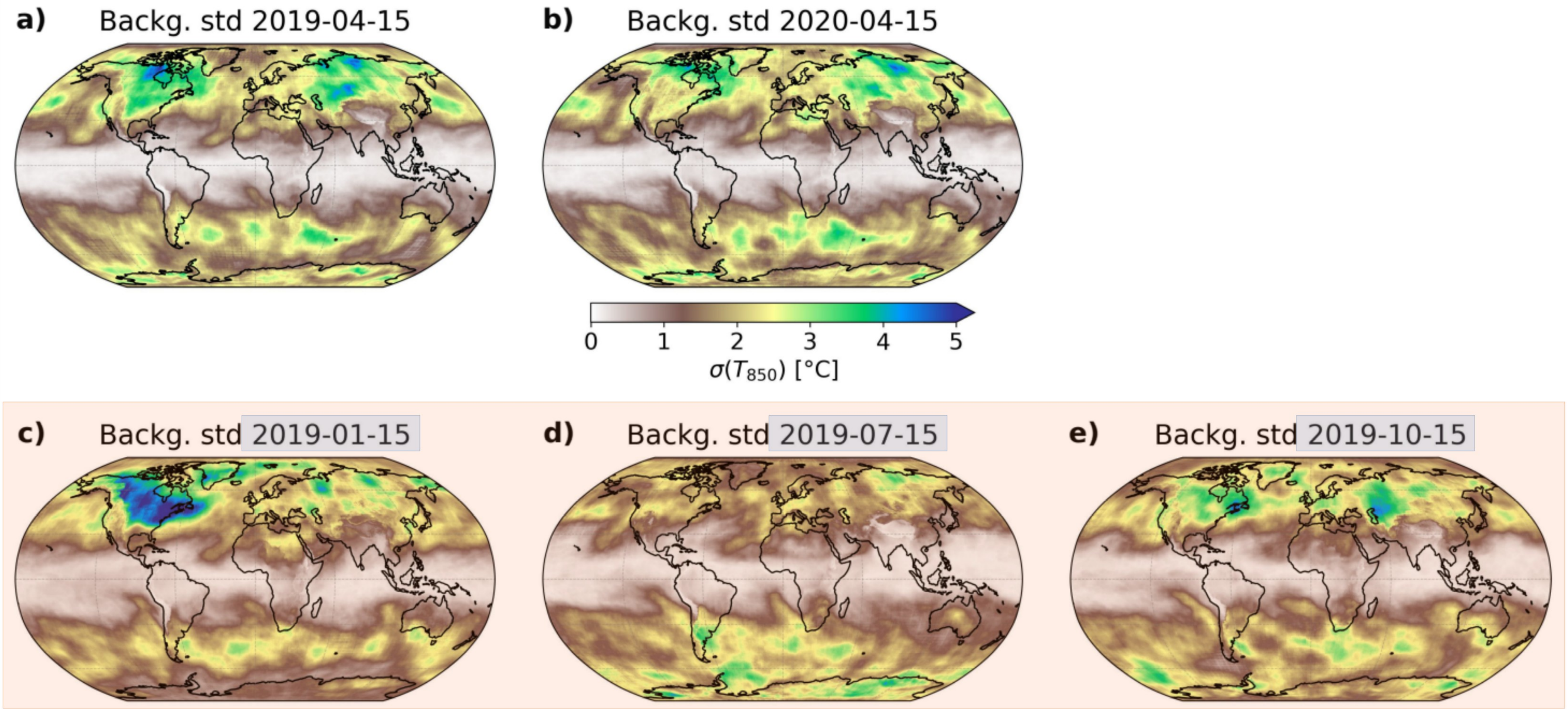
$$z_i = \mu_i + \hat{z}_i\,\sigma_{bi}, \quad \hat{z}_i \sim \mathcal{N}(0,1)$$



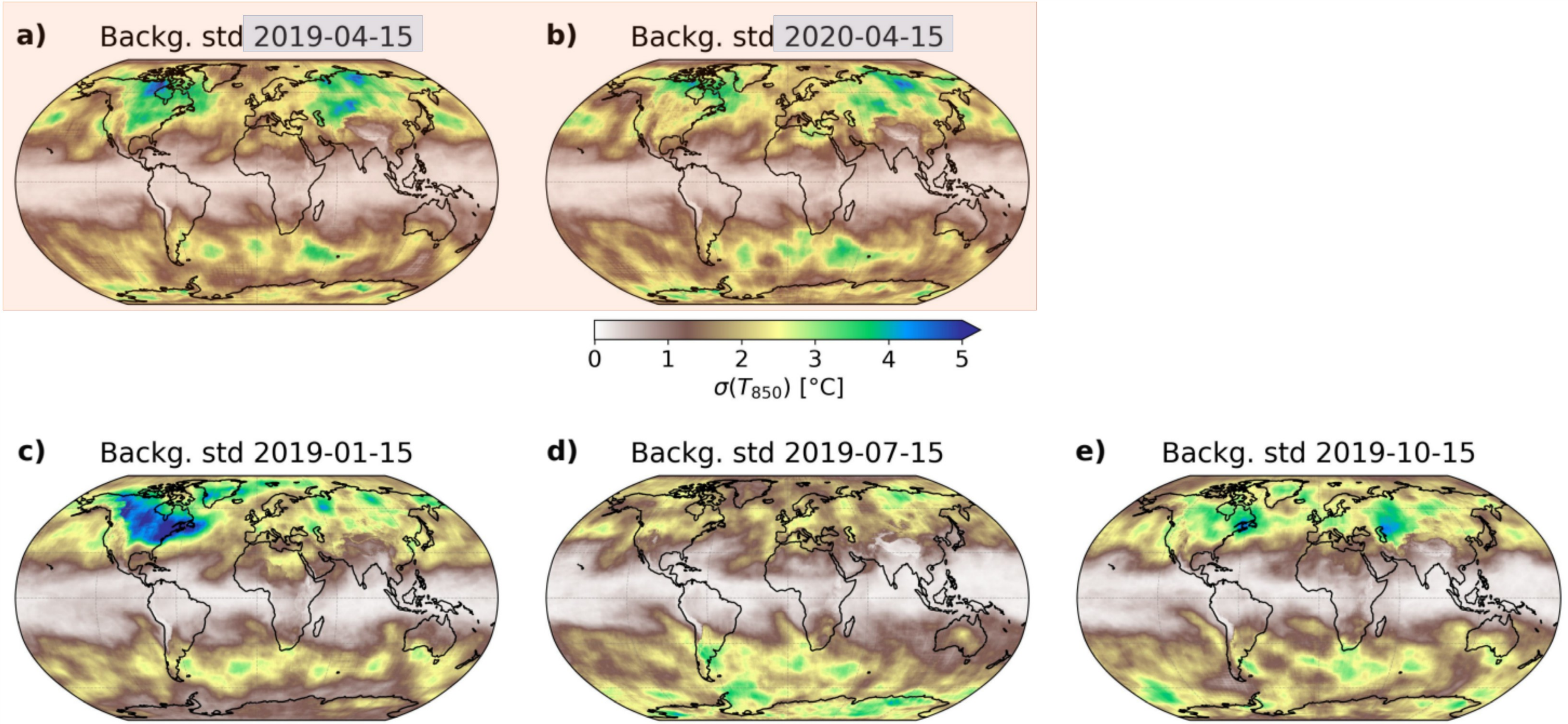
# Flow-dependent background-error standard deviation



# Flow-dependent background-error standard deviation

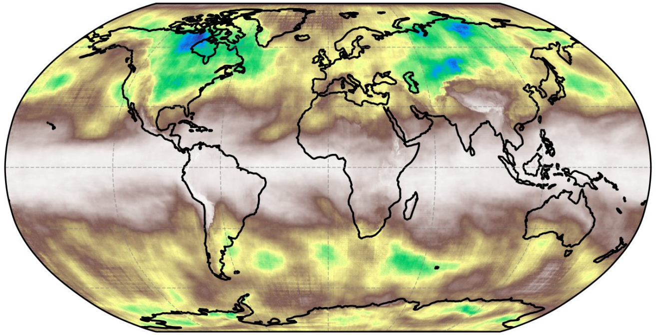


# Flow-dependent background-error standard deviation

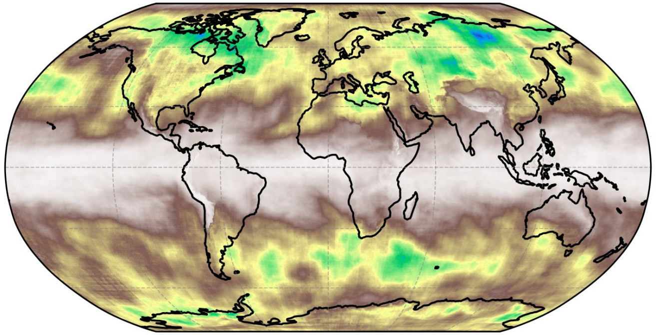


# Flow-dependent background-error standard deviation

a) Backg. std 2019-04-15

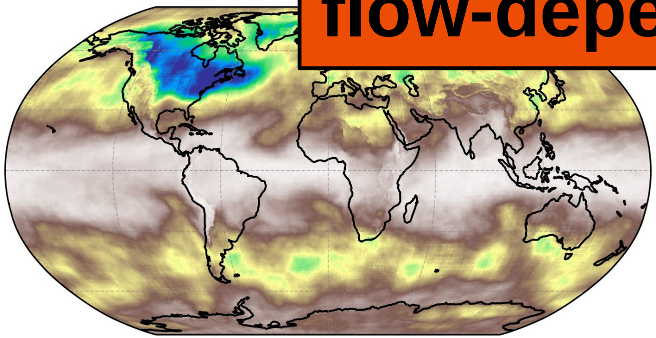


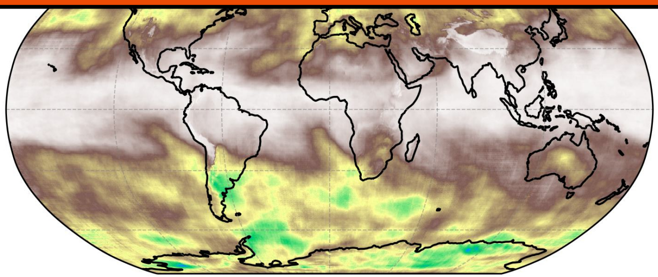
b) Backg. std 2020-04-15



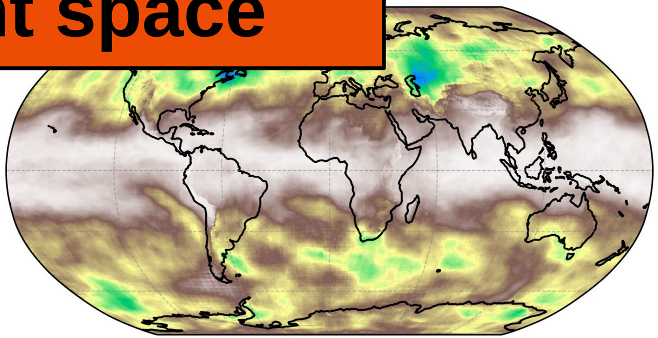
Even though background-error standard deviation is constant in the latent space, it is **flow-dependent in the grid point space**

c) Backg. st





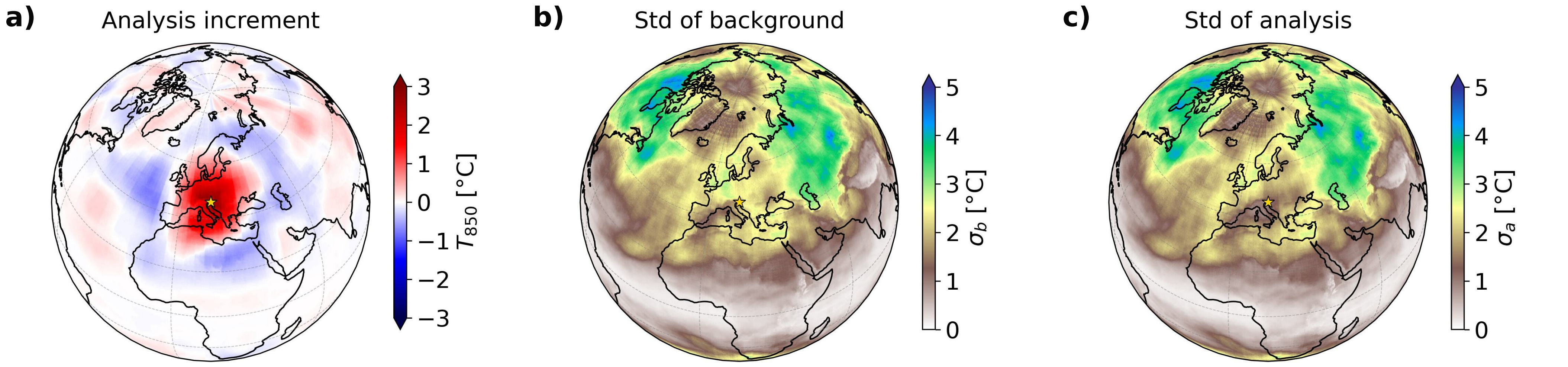
19-10-15

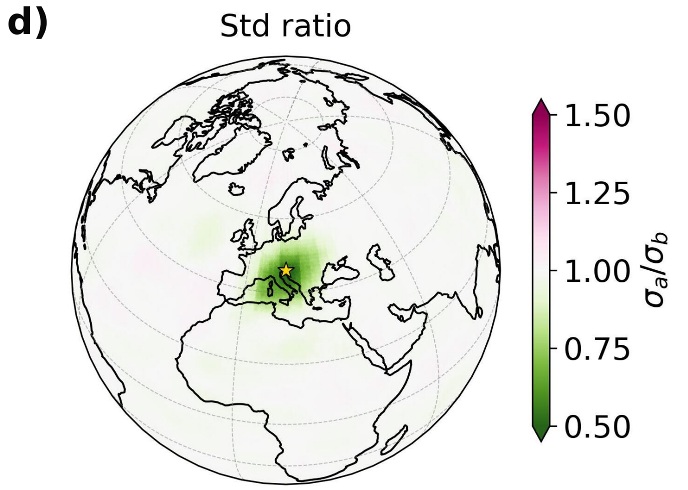


# Single observation experiments in midlatitudes

# Example: observation above **Ljubljana, Slovenia** (46.1˚N, 14.5˚E)

- Background for 2019-04-15
- Preset observation departure $\delta T^o_{850} = T^o_{850} - T^b_{850} = 3\,\mathrm{K}$ and standard deviation $\sigma_o = 1\,\mathrm{K}$



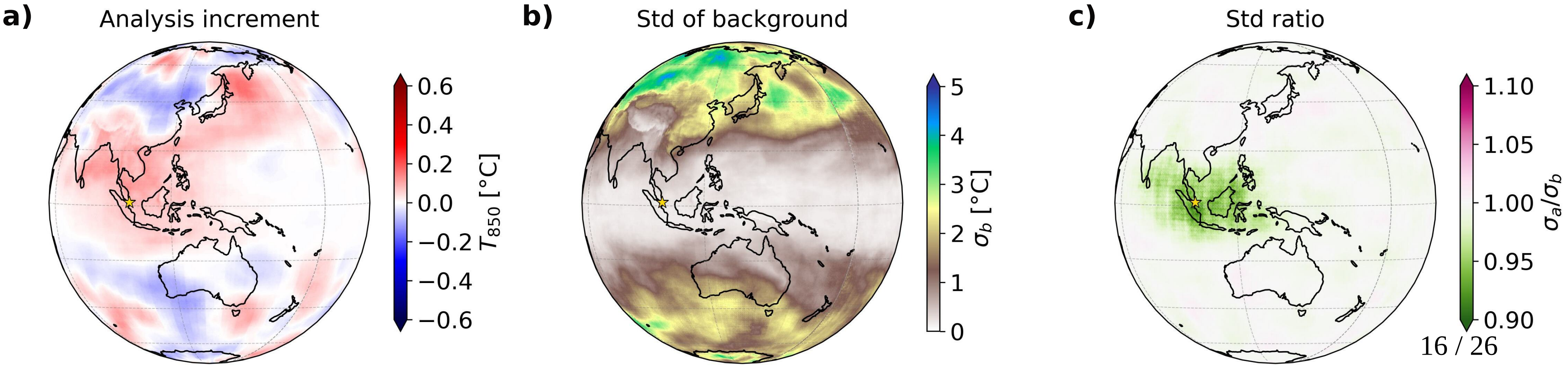




- Increment peaks at the observation location
- Increment stretched in SW-NE and elongated towards SW (typical SW winds)
- Positive increment surrounded by a shallower negative increment (spatial translation of synoptic Rossby waves typical for climatological **B** matrices (Fisher, 2003))
- Increments further away have negligible magnitude
- $\sigma_a$ significantly reduced with respect to $\sigma_b$ only in the area of the positive increment

# Single observation experiments in tropics

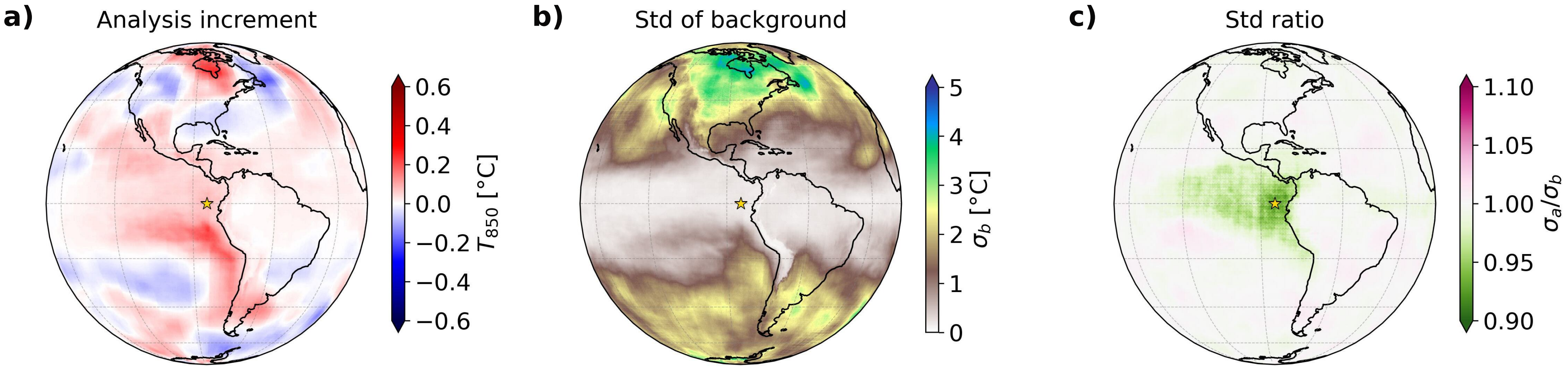
# Example: observation above **Singapore** (1.3˚N, 103.9˚E)

- Weak increment as $\sigma_o \gg \sigma_b$
- Same magnitude of increment in tropics and midlatitudes as $\sigma_b$ in the midlatitudes is much greater than in the tropics (climatological **B** matrix (Fisher, 2003))
- Std reduction elongated towards E (typical E winds in the tropical lower troposphere)
- Increment pattern resembles response to diabatic heating over the Maritime continent (Gill, 1980; Hoskins and Karoly, 1981)



# Example: observation above **E Equatorial Pacific** (0˚N, 85˚W)

- ENSO pattern
- $\sigma_a/\sigma_b$ reduction elongated towards W (lower branch of Pacific Walker circulation)



# Quantitative evaluation for single observation experiments

- Theoretical analysis increment and standard deviation at observation location:

$$\delta T^a_{850} = \frac{\delta T^o_{850}/\sigma_o^2}{1/\sigma_b^2 + 1/\sigma_o^2} \qquad \sigma_a = \sqrt{\frac{1}{1/\sigma_b^2 + 1/\sigma_o^2}}$$

- Experimental results:

| Location | $\boldsymbol{\delta T^o_{850}}$ | $\boldsymbol{\sigma_o}$ | $\boldsymbol{\sigma_b}$ | Theo. $\boldsymbol{\delta T^a_{850}}$ | Ex. $\boldsymbol{\delta T^a_{850}}$ | Theo. $\boldsymbol{\sigma_a}$ | Ex. $\boldsymbol{\sigma_a}$ |
|---|---|---|---|---|---|---|---|
| Ljubljana | 3.03 | 1.07 | 1.91 | 2.31 | 2.19 | 0.93 | 0.94 |
| SW Indian Ocean | 3.14 | 0.95 | 3.86 | 2.96 | 2.95 | 0.92 | 0.95 |
| Singapore | 3.11 | 0.99 | 0.19 | 0.11 | 0.08 | 0.18 | 0.18 |
| Equatorial Africa | 3.14 | 1.10 | 0.61 | 0.75 | 0.59 | 0.54 | 0.54 |
| E Pacific | 2.93 | 1.08 | 0.22 | 0.12 | 0.09 | 0.22 | 0.21 |

# Quantitative evaluation for single observation experiments

- Theoretical analysis increment and standard deviation at observation location:

$$\delta T^a_{850} = \frac{\delta T^o_{850}/\sigma_o^2}{1/\sigma_b^2 + 1/\sigma_o^2} \qquad \sigma_a = \sqrt{\frac{1}{1/\sigma_b^2 + 1/\sigma_o^2}}$$

- Experimental results:

| Location | $\boldsymbol{\delta T^o_{850}}$ | $\boldsymbol{\sigma_o}$ | $\boldsymbol{\sigma_b}$ | Theo. $\boldsymbol{\delta T^a_{850}}$ | Ex. $\boldsymbol{\delta T^a_{850}}$ | Theo. $\boldsymbol{\sigma_a}$ | Ex. $\boldsymbol{\sigma_a}$ |
|---|---|---|---|---|---|---|---|
| Ljubljana | 3.03 | 1.07 | 1.91 | 2.31 | 2.19 | 0.93 | 0.94 |
| SW Indian Ocean | 3.14 | 0.95 | 3.86 | 2.96 | 2.95 | 0.92 | 0.95 |
| Singapore | 3.11 | 0.99 | 0.19 | 0.11 | 0.08 | 0.18 | 0.18 |
| Equatorial Africa | 3.14 | 1.10 | 0.61 | 0.75 | 0.59 | 0.54 | 0.54 |
| E Pacific | 2.93 | 1.08 | 0.22 | 0.12 | 0.09 | 0.22 | 0.21 |

# Quantitative evaluation for single observation experiments

- Theoretical analysis increment and standard deviation at observation location:

$$\delta T^a_{850} = \frac{\delta T^o_{850}/\sigma_o^2}{1/\sigma_b^2 + 1/\sigma_o^2} \qquad \sigma_a = \sqrt{\frac{1}{1/\sigma_b^2 + 1/\sigma_o^2}}$$

- Experimental results:

| Location | $\boldsymbol{\delta T^o_{850}}$ | $\boldsymbol{\sigma_o}$ | $\boldsymbol{\sigma_b}$ | Theo. $\boldsymbol{\delta T^a_{850}}$ | Ex. $\boldsymbol{\delta T^a_{850}}$ | Theo. $\boldsymbol{\sigma_a}$ | Ex. $\boldsymbol{\sigma_a}$ |
|---|---|---|---|---|---|---|---|
| Ljubljana | 3.03 | 1.07 | 1.91 | 2.31 | 2.19 | 0.93 | 0.94 |
| SW Indian Ocean | 3.14 | 0.95 | 3.86 | 2.96 | 2.95 | 0.92 | 0.95 |
| Singapore | 3.11 | 0.99 | 0.19 | 0.11 | 0.08 | 0.18 | 0.18 |
| Equatorial Africa | 3.14 | 1.10 | 0.61 | 0.75 | 0.59 | 0.54 | 0.54 |
| E Pacific | 2.93 | 1.08 | 0.22 | 0.12 | 0.09 | 0.22 | 0.21 |

arXiv > physics > arXiv:2308.16073

Search...

Help | Adva

Physics > Atmospheric and Oceanic Physics

*[Submitted on 30 Aug 2023]*

# Neural-Network Data Assimilation using Variational Autoencoder

Boštjan Melinc, Žiga Zaplotnik

In numerical weather prediction, data assimilation of atmospheric observations traditionally relies on variational and Kalman filter methods. Here, we propose an alternative full neural-network data assimilation (NNDA) in the latent space with variational autoencoder (VAE). The 3D variational data assimilation (3D-Var) cost function is applied to find the latent space vector which optimally fuses simulated observations and the encoded short-range persistence forecast (background), accounting for their errors. We demonstrate that the background-error covariance matrix, measured and represented in the latent space, is quasi-diagonal. Data assimilation experiments with a single temperature observation in the lower troposphere indicate that the same set of neural-network-derived basis functions is able to describe both tropical and extratropical background-error covariances. The background-error covariances evolve seasonally and also depend on the current state of the atmosphere. Our method mimics the 3D variational data assimilation (3D-Var), however, it can be further extended to resemble 4D-Var by including the neural network forecast model.

Comments: 23 pages, 17 figures. Submitted to QJRMS

Subjects: **Atmospheric and Oceanic Physics (physics.ao-ph)**

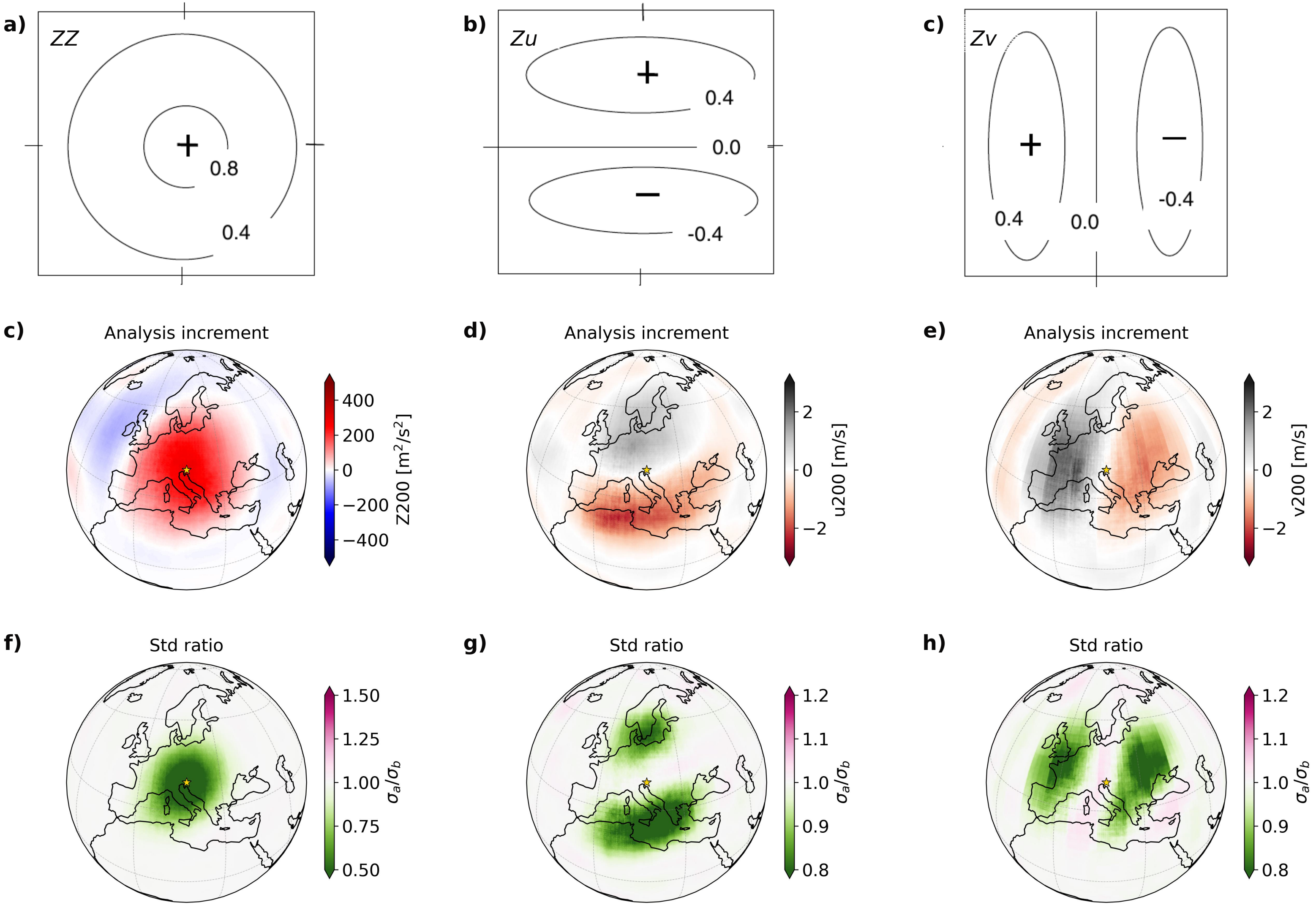
Cite as: arXiv:2308.16073 **[physics.ao-ph]**

(or arXiv:2308.16073v1 **[physics.ao-ph]** for this version)

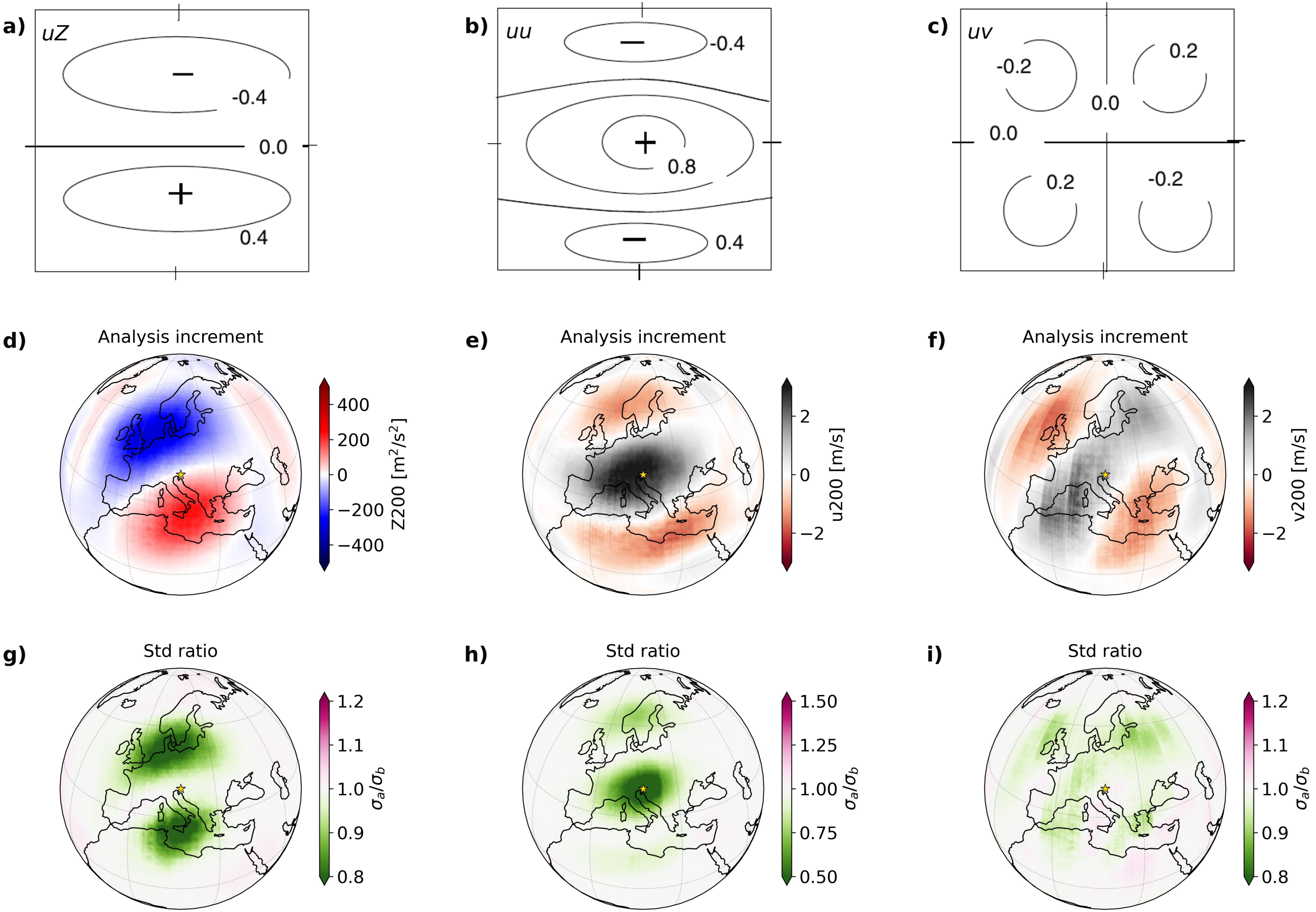
https://doi.org/10.48550/arXiv.2308.16073

# Multivariate case: $(Z_{200}, u_{200}, v_{200})$

- **Ljubljana**
- **Observed** $Z_{200}$
- $\delta Z^o_{200} = 300\,\mathrm{m}^2/\mathrm{s}^2$
- $\sigma_o = 100\,\mathrm{m}^2/\mathrm{s}^2$
- Top row: Correlation and cross-correlation functions derived using the geostrophic increment assumption (from Kalnay, 2003)



- **Ljubljana**
- **Observed** $u_{200}$
- $\delta u^o_{200} = 3\,\mathrm{m/s}$
- $\sigma_o = 1\,\mathrm{m/s}$



# Outlook and conclusions

# Outlook

- Multivariate case with other variables (humidity, mean sea level pressure, etc.)
- More than one pressure/hybrid level
- 4D-Var
- Flow-dependent $\mathbf{B}_z$ using ensemble of forecast models simulations

- Potential pitfalls:
  - Should we localise increments?
  - Can we use the same approach for mesoscale/convective-scale balances?

# Conclusions

- We propose a neural-network-based method for variational data assimilation of atmospheric observations in a reduced-dimension latent space discovered by VAE
- We define a 3D-Var cost function in the latent space
- $\mathbf{B}_z$ is shown to be **quasi-diagonal**
- $\mathbf{B}_z$ provides a **unified representation of both tropical and extratropical covariances**
- $\mathbf{B}_z$ is constant in the latent space but **flow dependent in the grid point space**
- The method can be further extended to multiple variables, multiple levels, and 4D-Var

# Bibliography


- 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. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12746 LNCS:373–386, 2021.
- P. Brohan. Machine Learning for Data Asimilation, 4 2022. URL https://github.com/philip-brohan/Proxy_20CR.
- M. Fisher. Background error covariance modelling. In *Seminar on Recent Development in Data Assimilation*, pages 45–63, Shinfield Park, Reading, 2003. ECMWF, ECMWF.
- A. E. Gill. Some simple solutions for heat-induced tropical circulation. *Quarterly Journal of the Royal Meteorological Society*, 106(449):3759–3777, 1980.
- B. J. Hoskins and D. J. Karoly. The steady linear response of a spherical atmosphere to thermal and orographic forcing. *Journal of the atmospheric sciences*, 38(6), 1179-1196, 1981.
- E. Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge Univesity Press, 2003.
- J. Mack, R. Arcucci, M. Molina-Solana, and Y. K. Guo. Attention-based Convolutional Autoencoders for 3D-Variational Data Assimilation. *Computer Methods in Applied Mechanics and Engineering*, 372:113291, 2020.
- B. Melinc and Z. Zaplotnik. Neural-network data assimilation using variational autoencoder. 2023. URL https://arxiv.org/abs/2308.16073.
- 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.