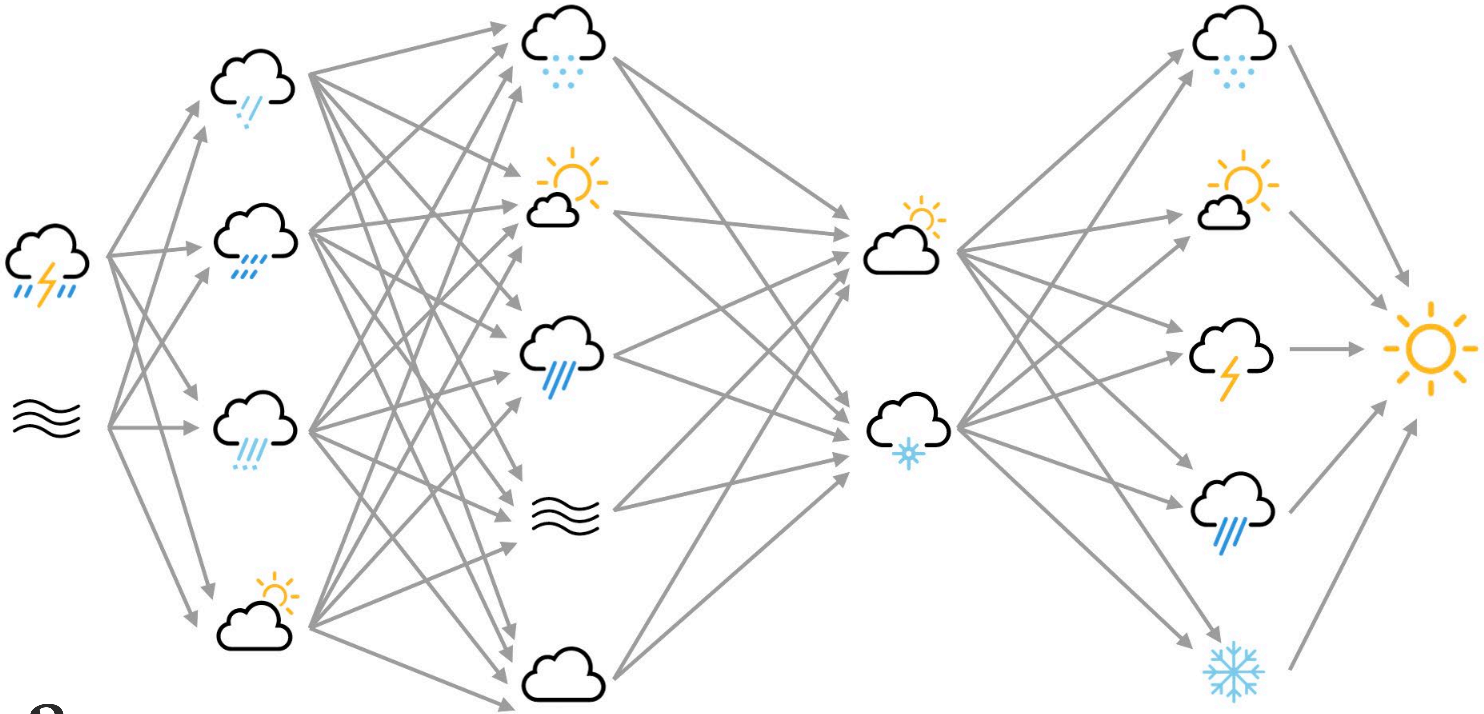


Data Assimilation with a Deep Convolutional Variational AutoEncoder

Easier, more flexible, and much, *much,* faster

Philip Brohan (philip.brohan@metoffice.gov.uk)

2022-11-15

https://thispersondoesnotexist.com/

https://thispersondoesnotexist.com/

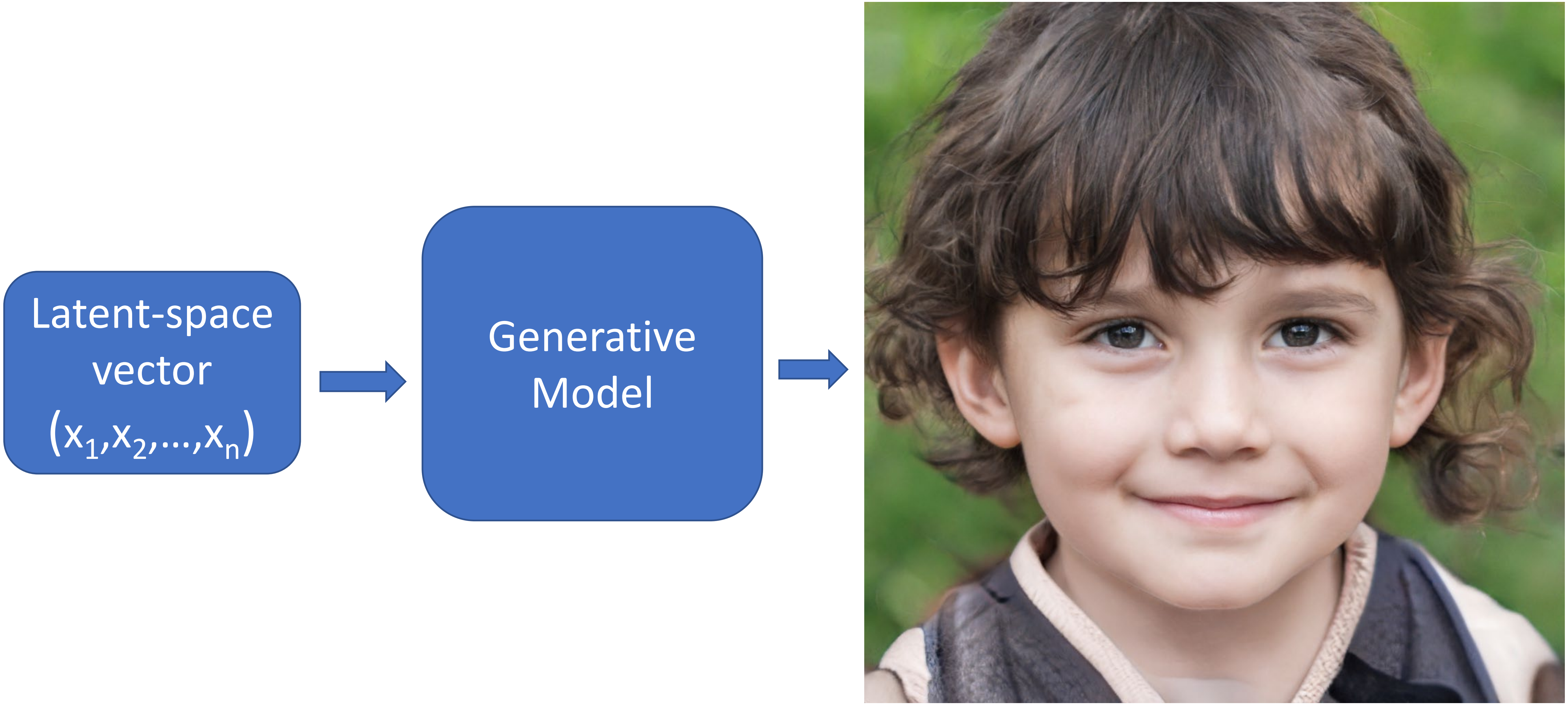
https://thispersondoesnotexist.com/

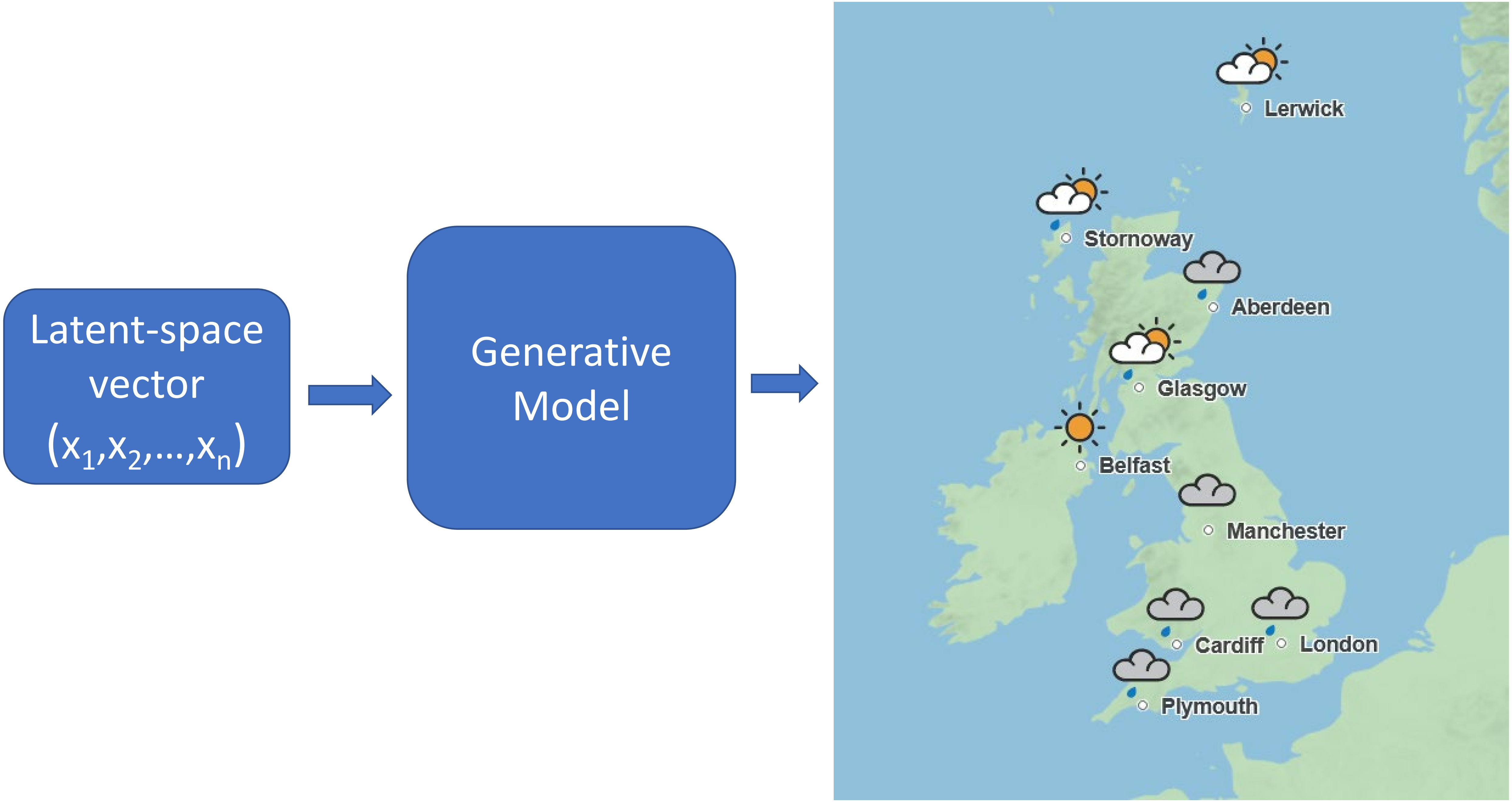
“A photograph of an astronaut riding a horse”

https://en.wikipedia.org/wiki/Stable_Diffusion

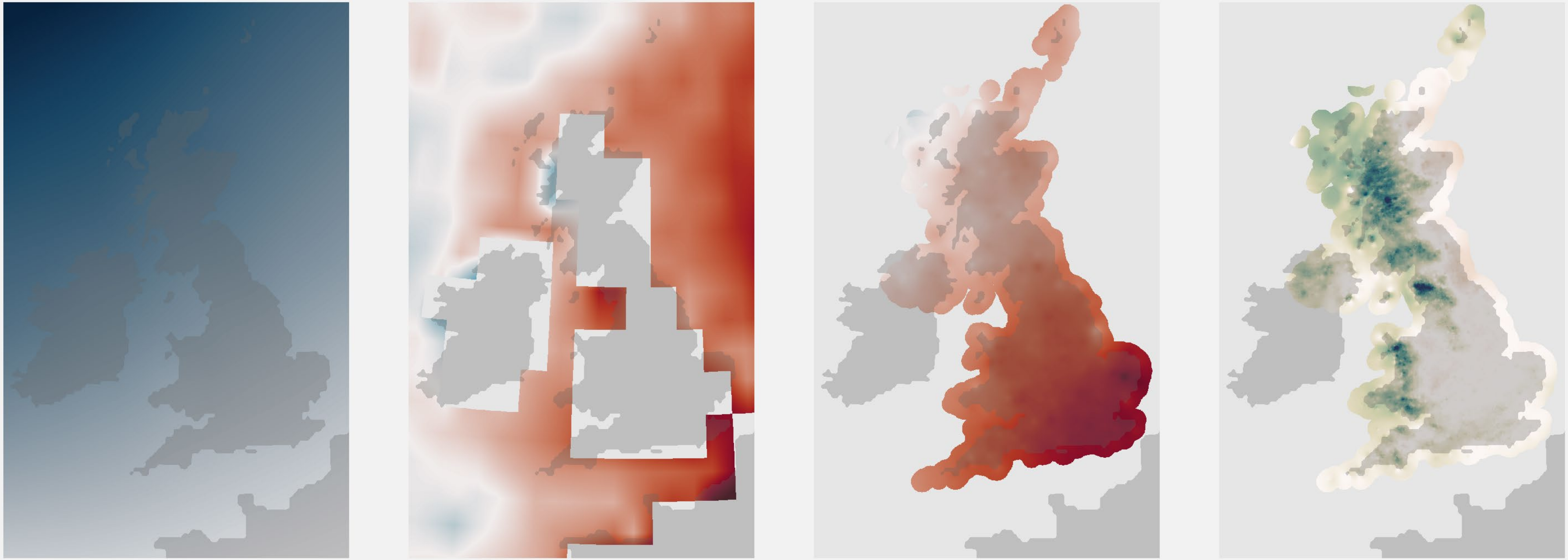
Hey, DALL-E, show me a “Scientist doing magic”

https://openai.com/blog/dall-e/


Latent-space vector $(x_1,x_2,\ldots,x_n)$
Generative Model


Latent-space vector $(x_1,x_2,\ldots,x_n)$
Generative Model
Lerwick
Stornoway
Aberdeen
Glasgow
Belfast
Manchester
Cardiff
London
Plymouth

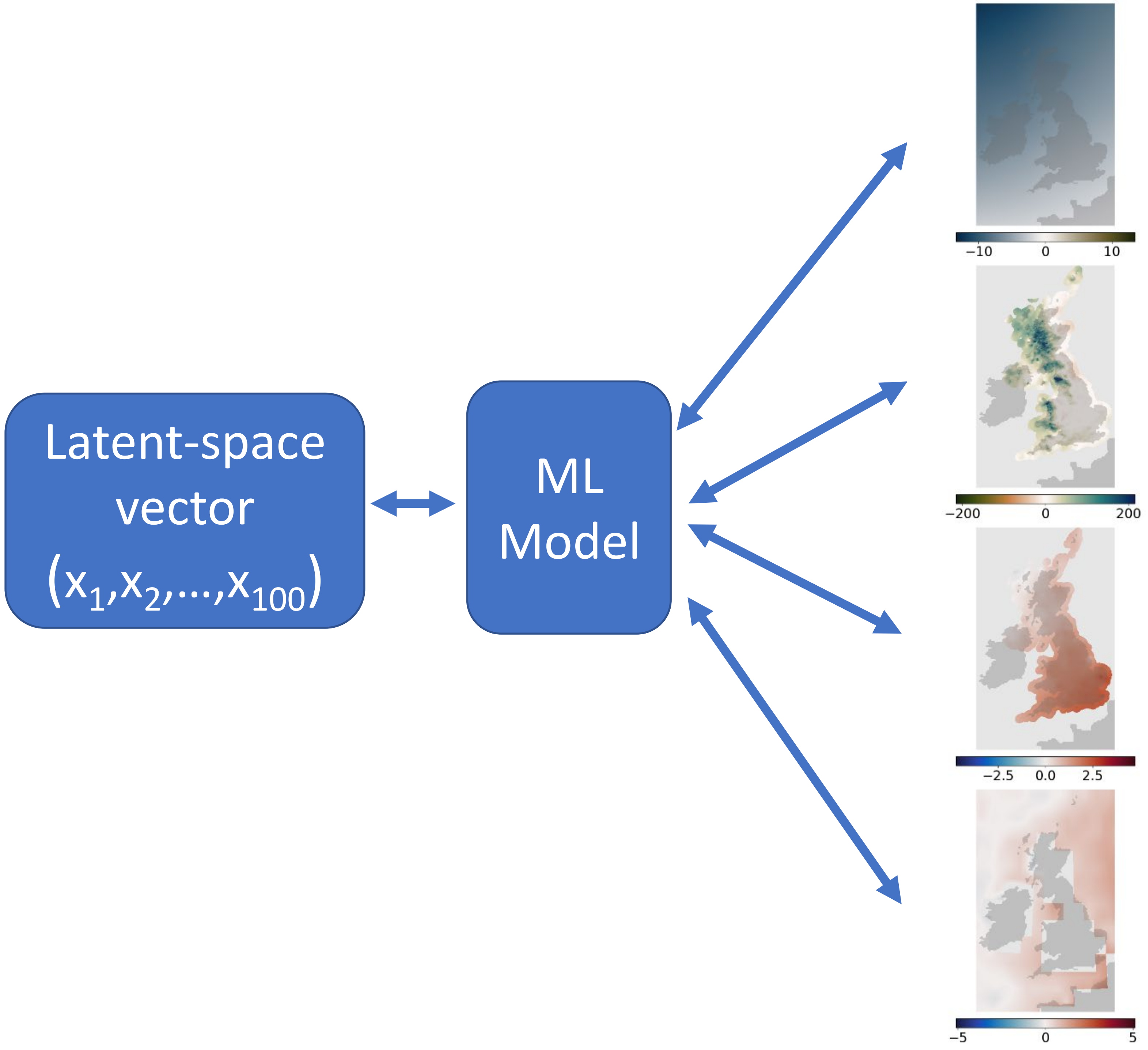
Four-variable state vector – monthly anomalies of near-surface weather:

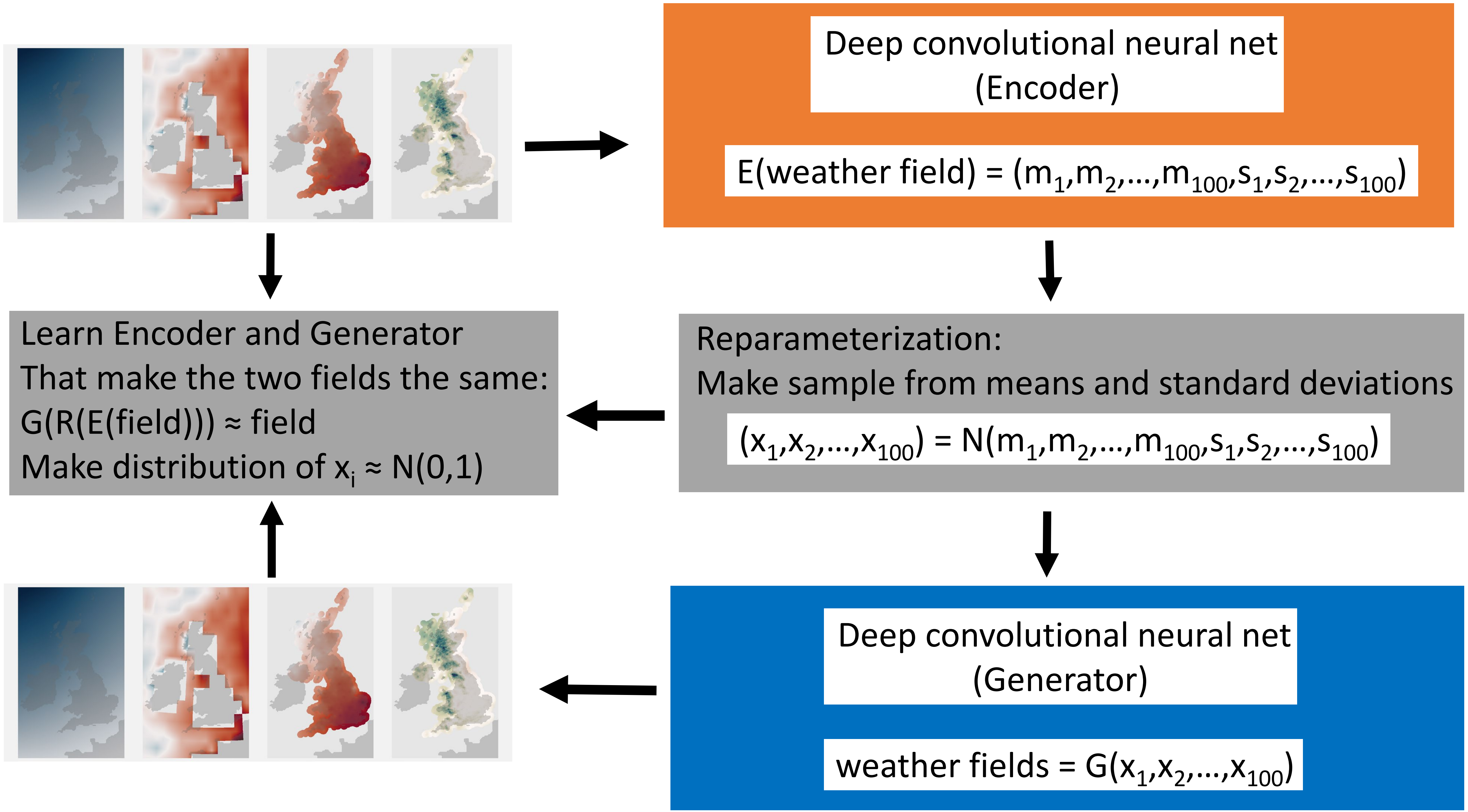


Pressure | SST | Air temperature | Precipitation

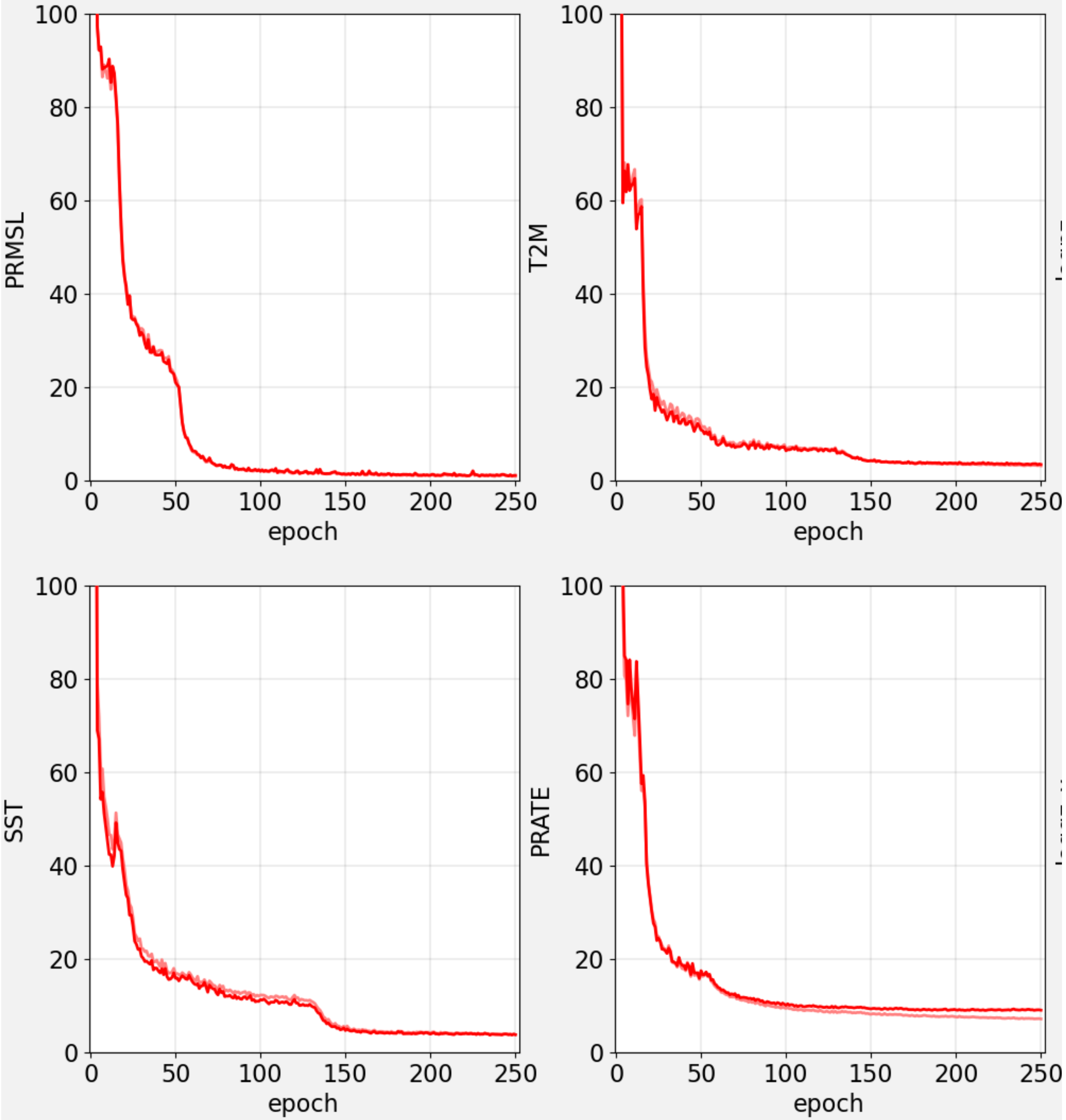
Air temperature and Precipitation from HadUK-Grid, SST and Pressure from 20CRv3.

March 1989


Latent-space vector
$(x_1,x_2,\ldots,x_{100})$
ML Model
−10 0 10
−200 0 200
−2.5 0.0 2.5
−5 0 5



Deep convolutional neural net
(Encoder)
E(weather field) = (m1,m2,...,m100,s1,s2,...,s100)
Learn Encoder and Generator
That make the two fields the same:
G(R(E(field))) ≈ field
Make distribution of xi ≈ N(0,1)
Reparameterization:
Make sample from means and standard deviations
(x1,x2,...,x100) = N(m1,m2,...,m100,s1,s2,...,s100)
Deep convolutional neural net
(Generator)
weather fields = G(x1,x2,...,x100)

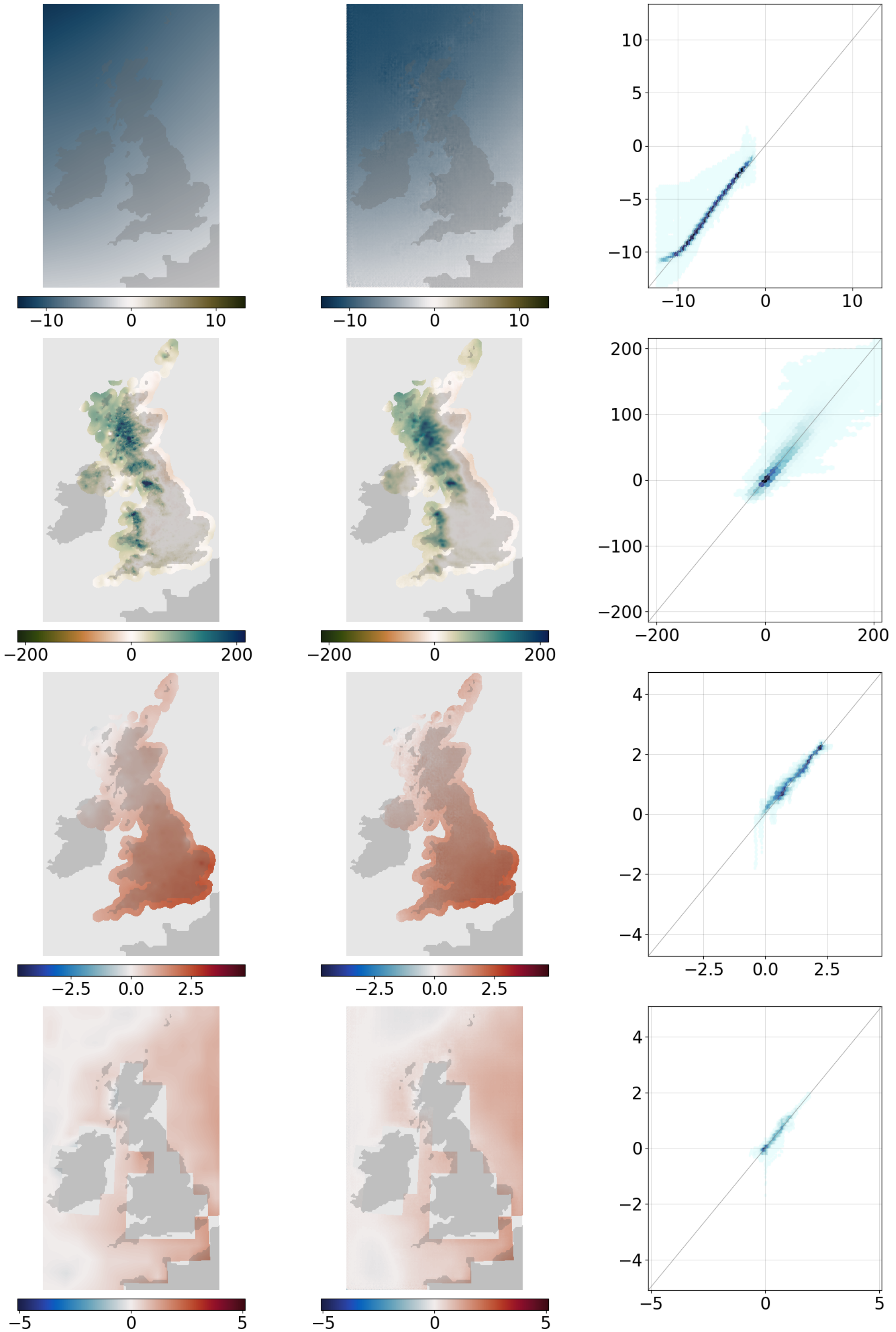

PRMSL
T2M
SST
PRATE
epoch
100
80
60
40
20
0
0
50
100
150
200
250

Validation for one test month (1989-03).

Left: Target (truth)

Middle: ML model output

Right: Target (x) v. model (y)

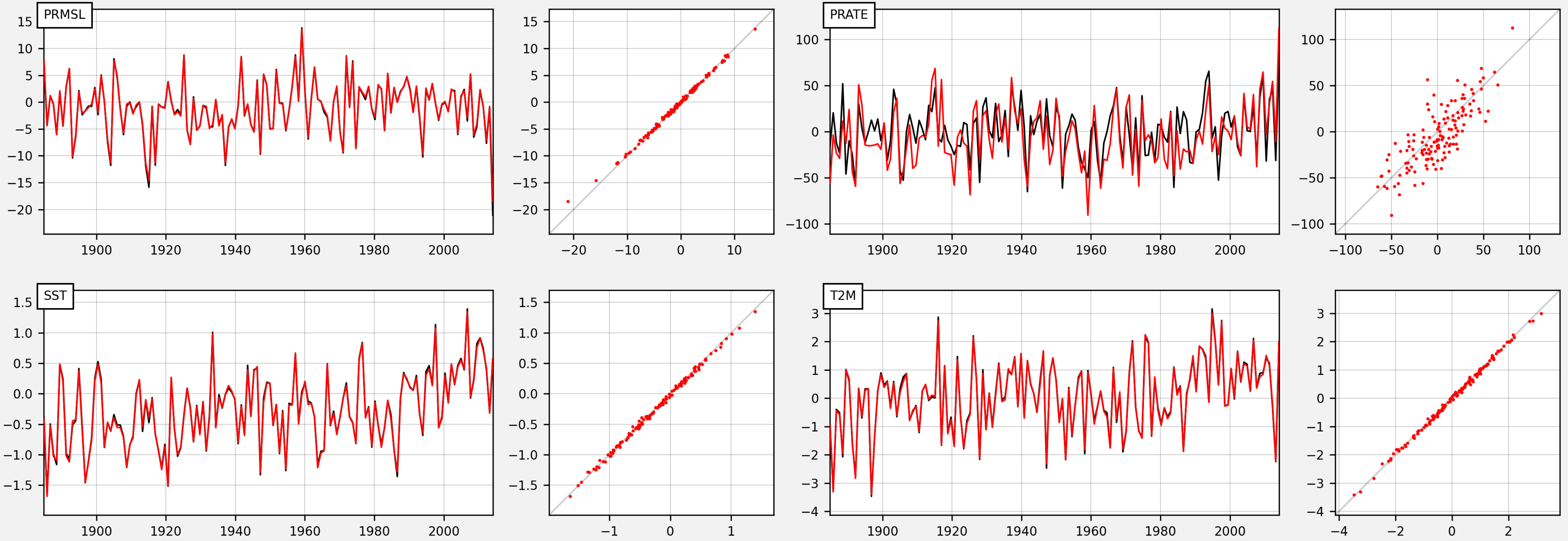


Pressure

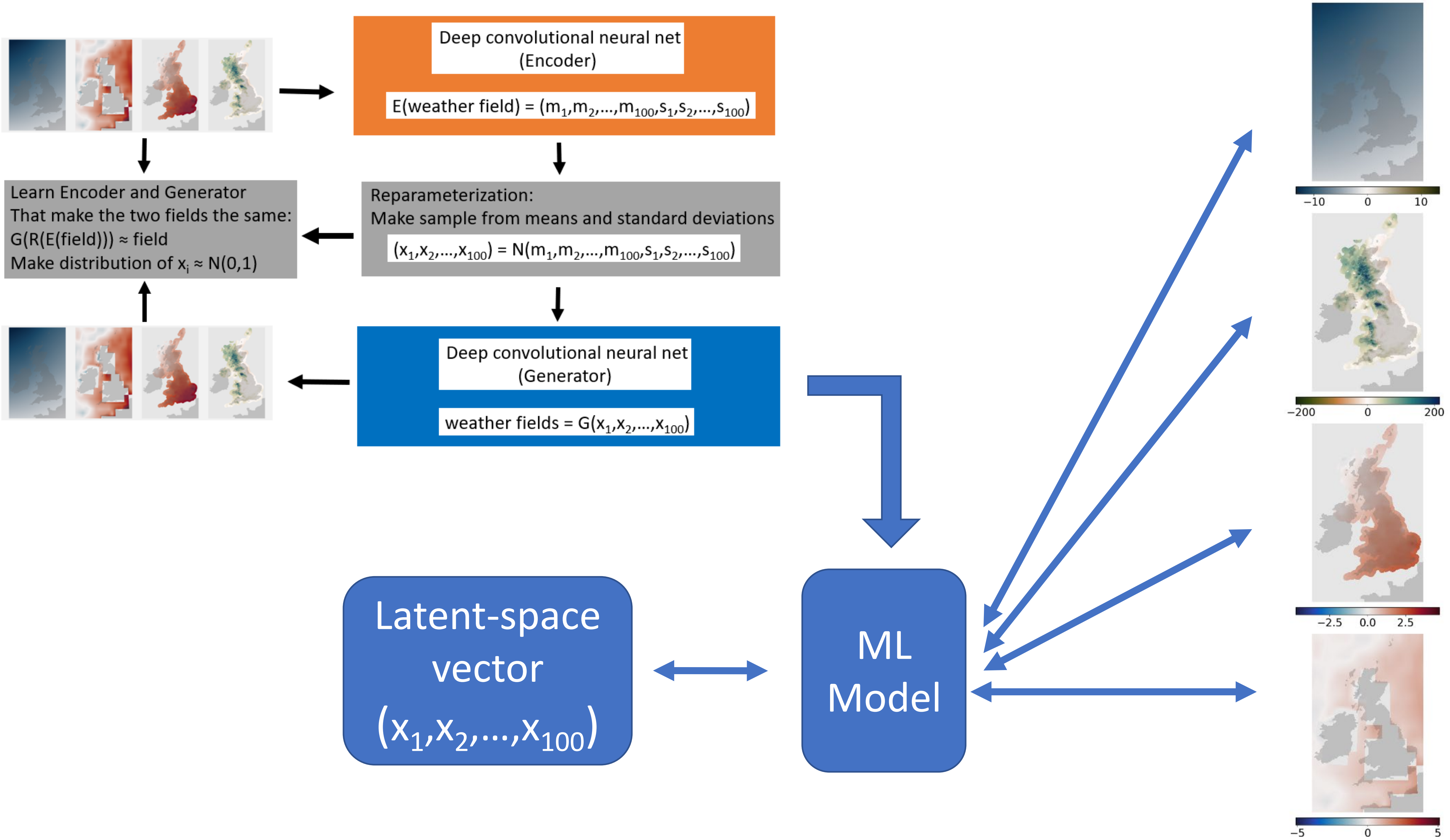
Precipitation

Air temperature

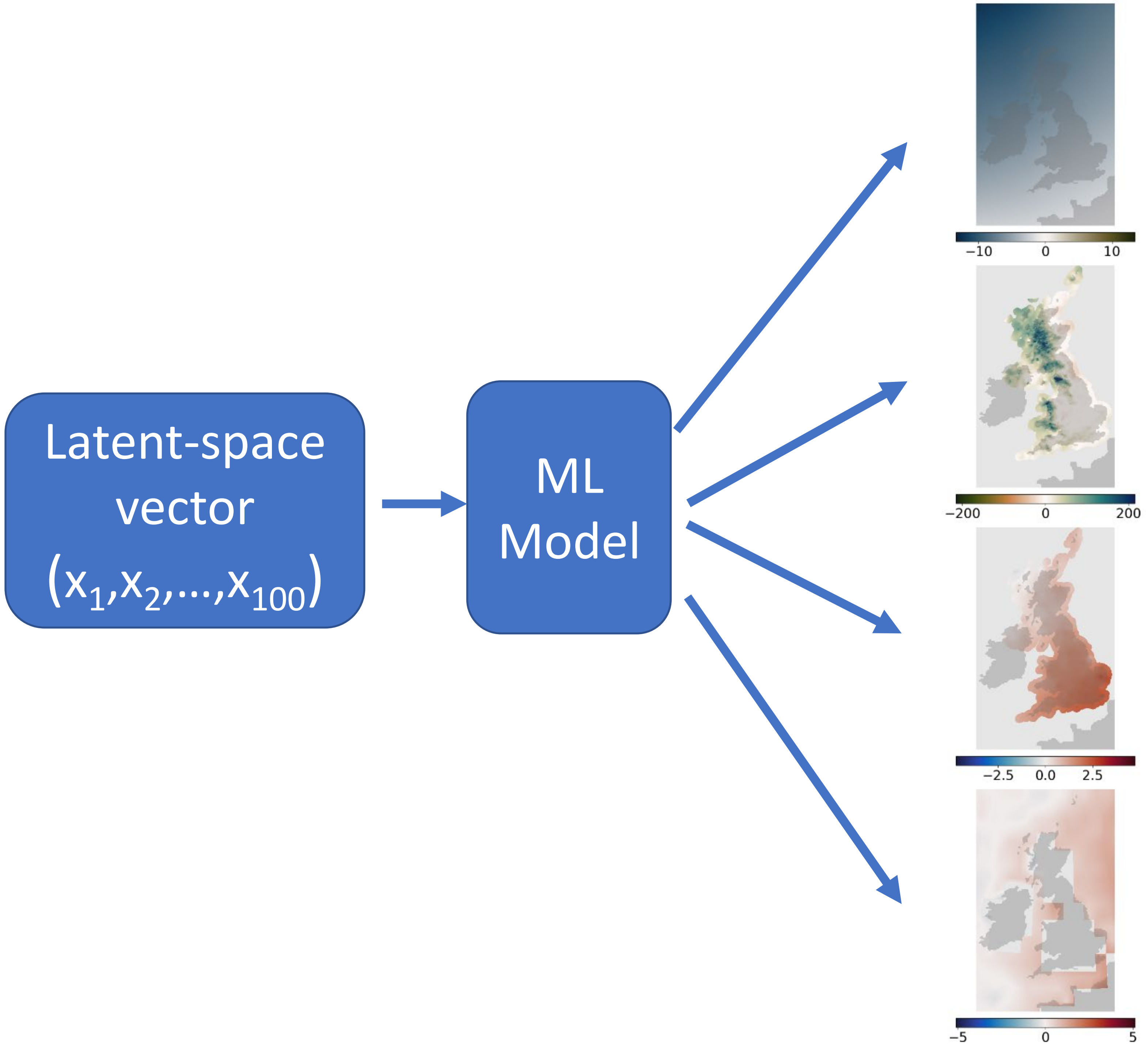
SST

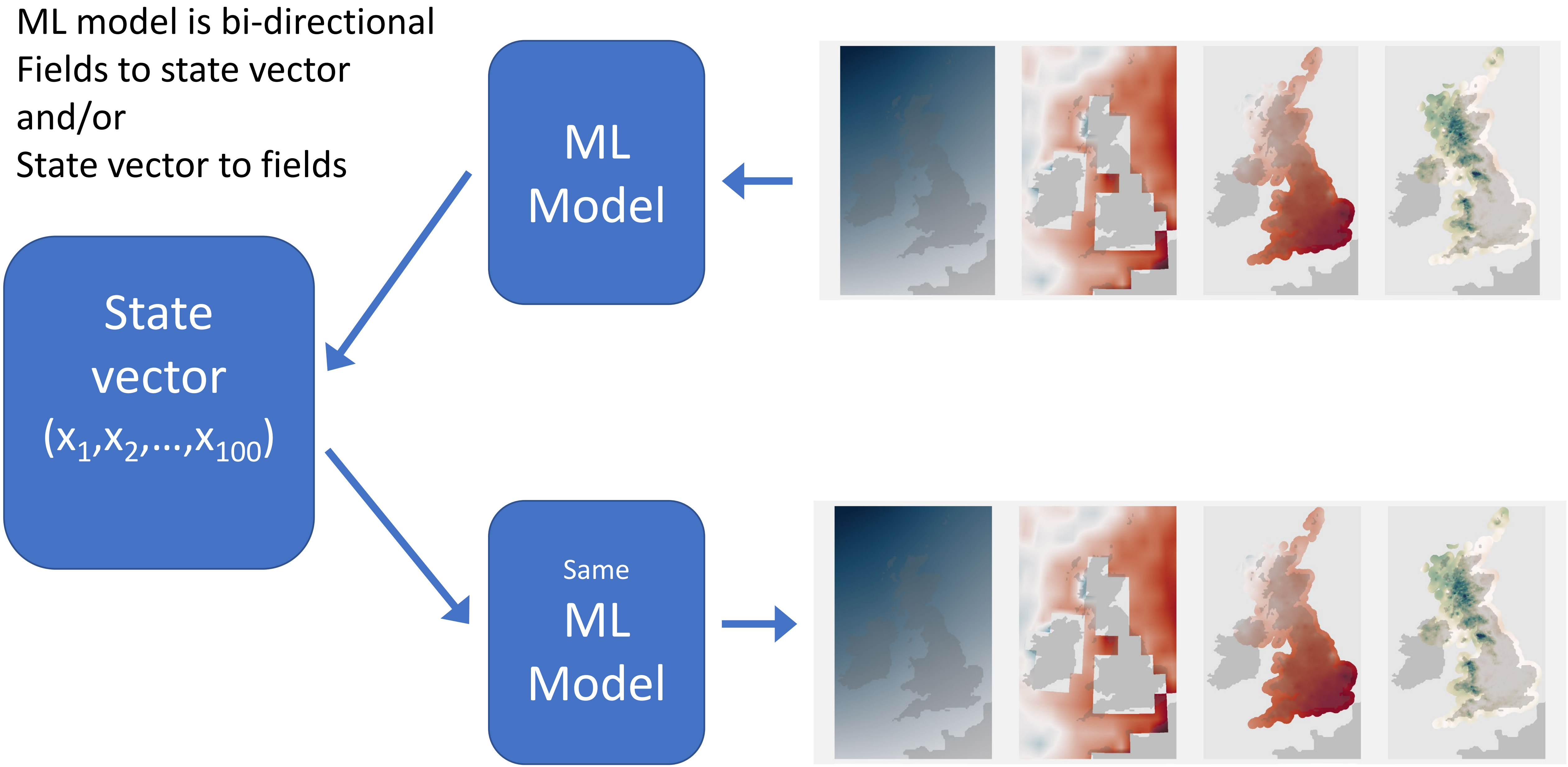


Validation for all the test months. Black – target, red – model. Means over the whole reconstructed field.



Deep convolutional neural net (Encoder)
E(weather field) = (m1,m2,…,m100,s1,s2,…,s100)
Reparameterization:
Make sample from means and standard deviations
(x1,x2,…,x100) = N(m1,m2,…,m100,s1,s2,…,s100)
Learn Encoder and Generator
That make the two fields the same:
G(R(E(field))) ≈ field
Make distribution of xi ≈ N(0,1)
Deep convolutional neural net (Generator)
weather fields = G(x1,x2,…,x100)
Latent-space vector (x1,x2,…,x100)
ML Model
−10 0 10
−200 0 200
−2.5 0.0 2.5
−5 0 5


Latent-space vector
$(x_1, x_2, \ldots, x_{100})$
ML Model
−10
0
10
−200
0
200
−2.5
0.0
2.5
−5
0
5

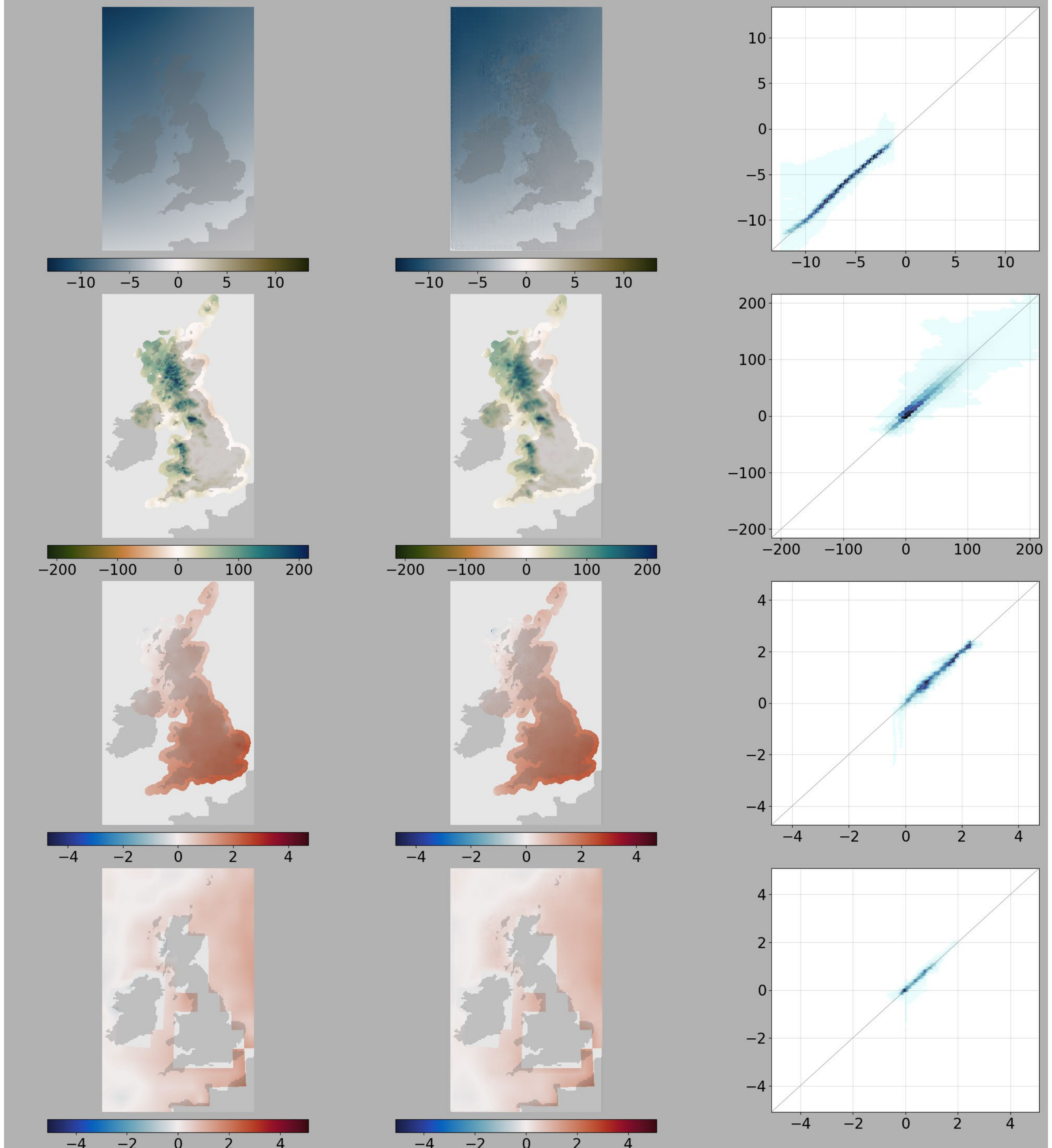

ML model is bi-directional
Fields to state vector
and/or
State vector to fields
ML Model
State vector $(x_1,x_2,\ldots,x_{100})$
Same
ML Model

All-fields assimilation for one test month (1989-03).

Left: Target (truth)

Middle: ML model output

Right: Target (x) v. model (y)

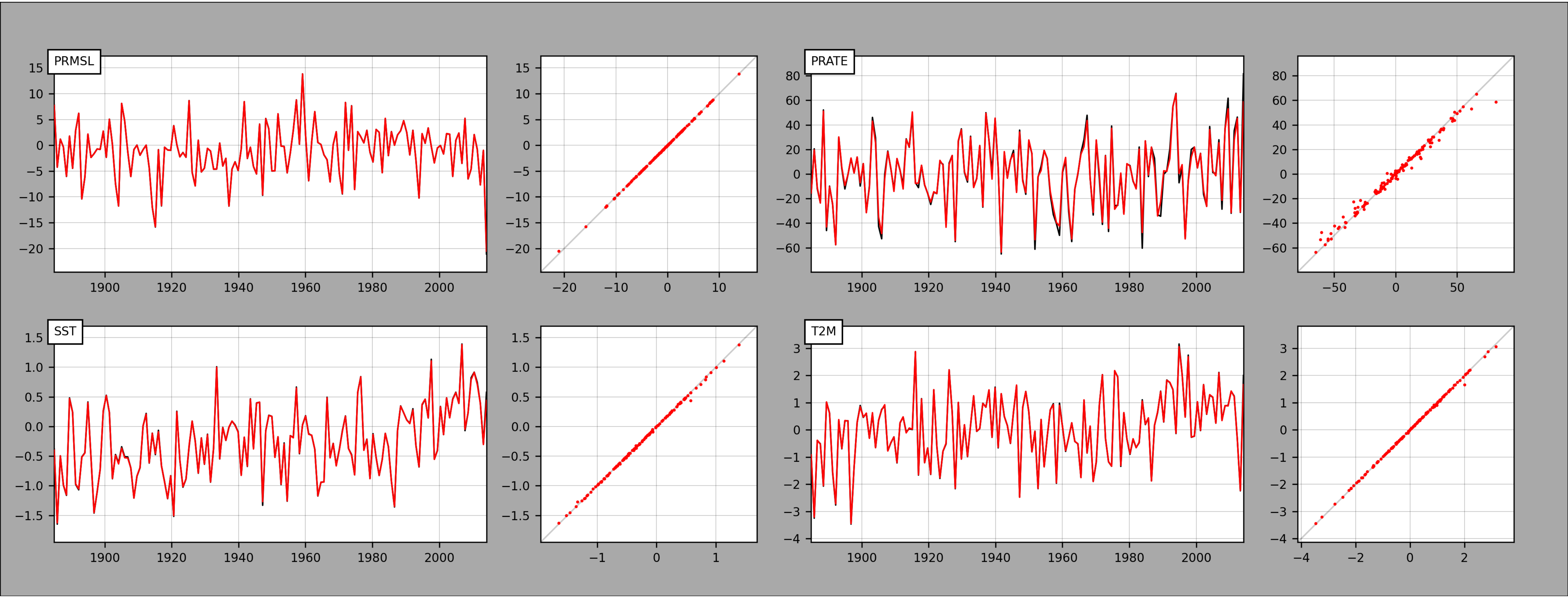


Pressure

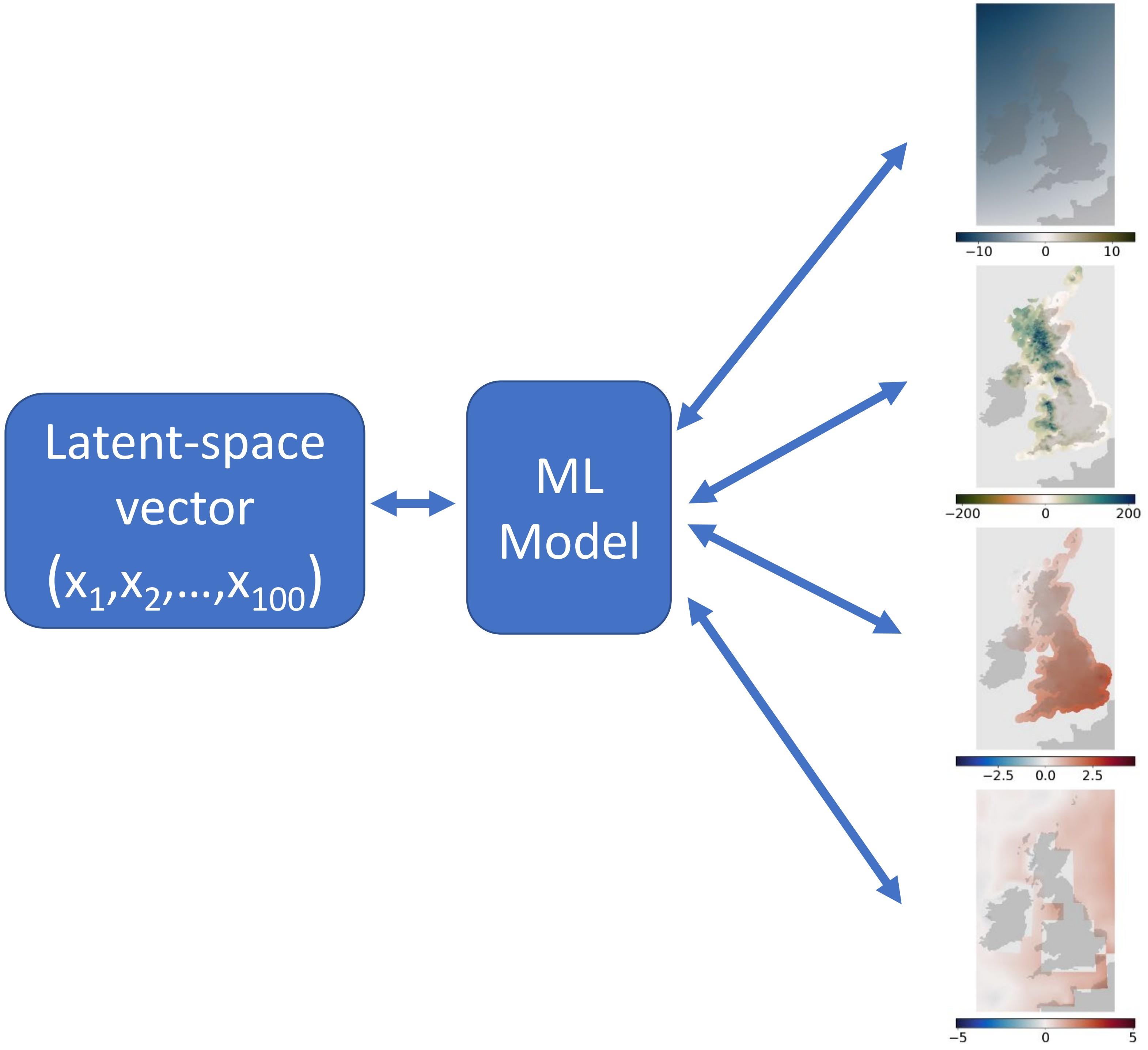
Precipitation

Air temperature

SST

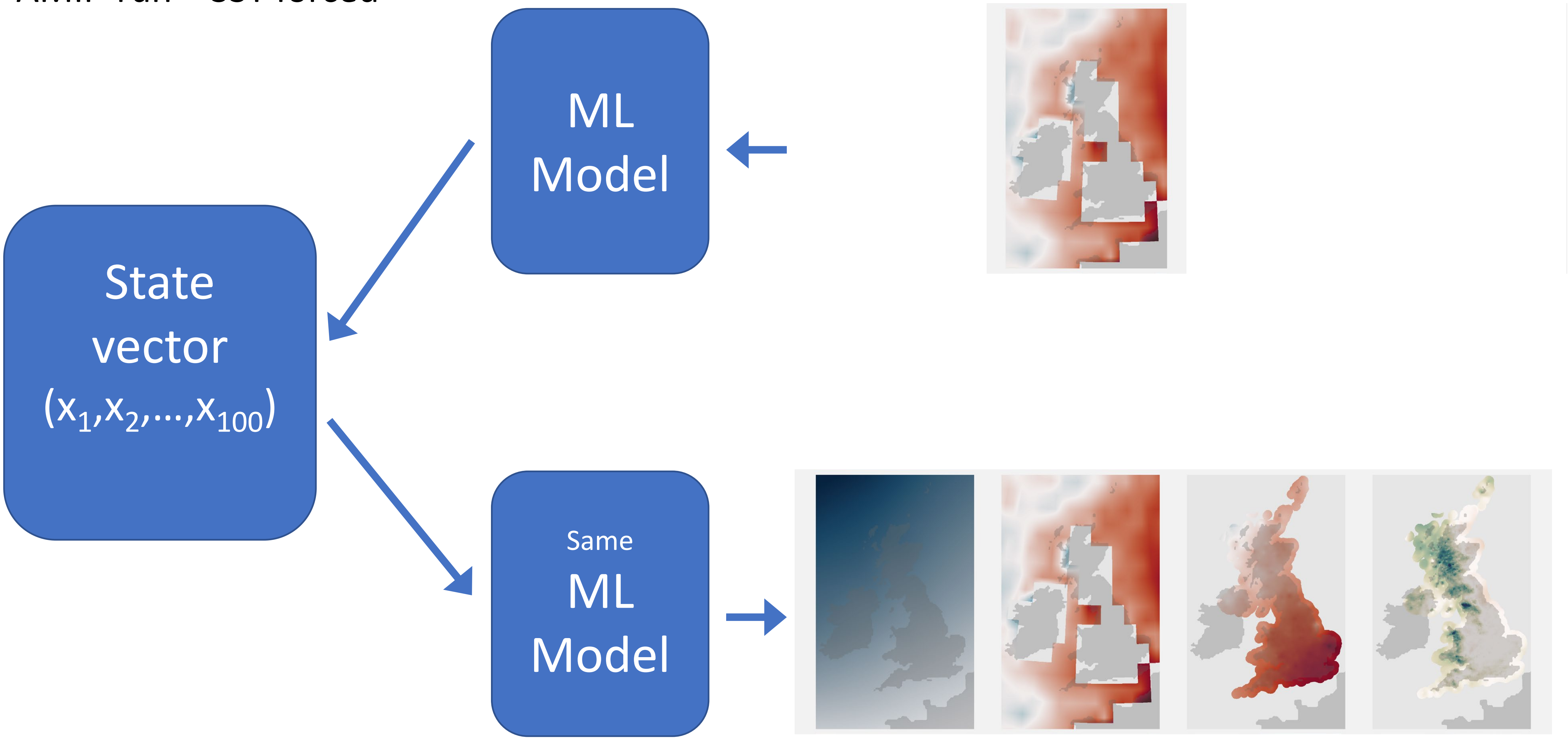


Results of assimilating all four fields. Black – target, red – model. Means over the whole reconstructed field.



Latent-space vector
$(x_1, x_2, \ldots, x_{100})$
ML Model
−10 0 10
−200 0 200
−2.5 0.0 2.5
−5 0 5

AMIP-run – SST forced

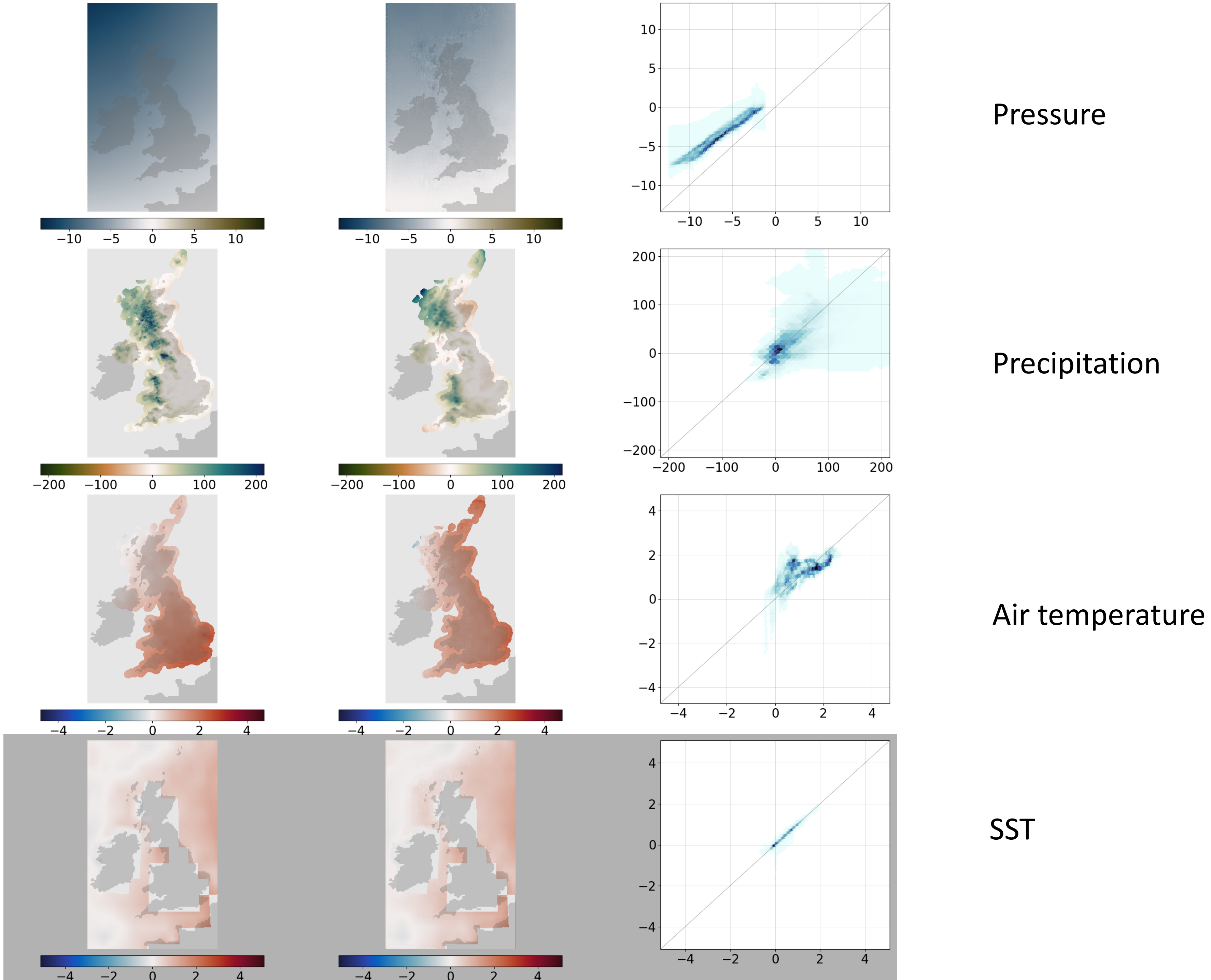


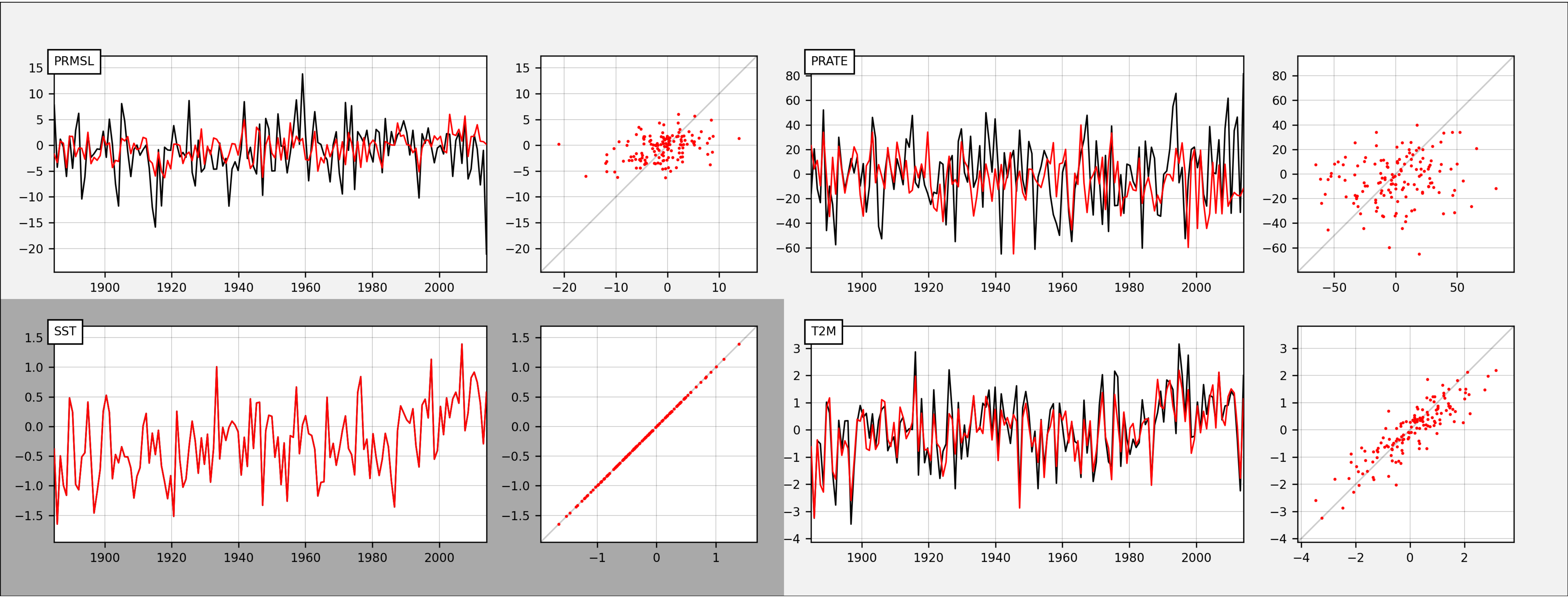
SST-only assimilation for one test month (1989-03).

Left: Target (truth)

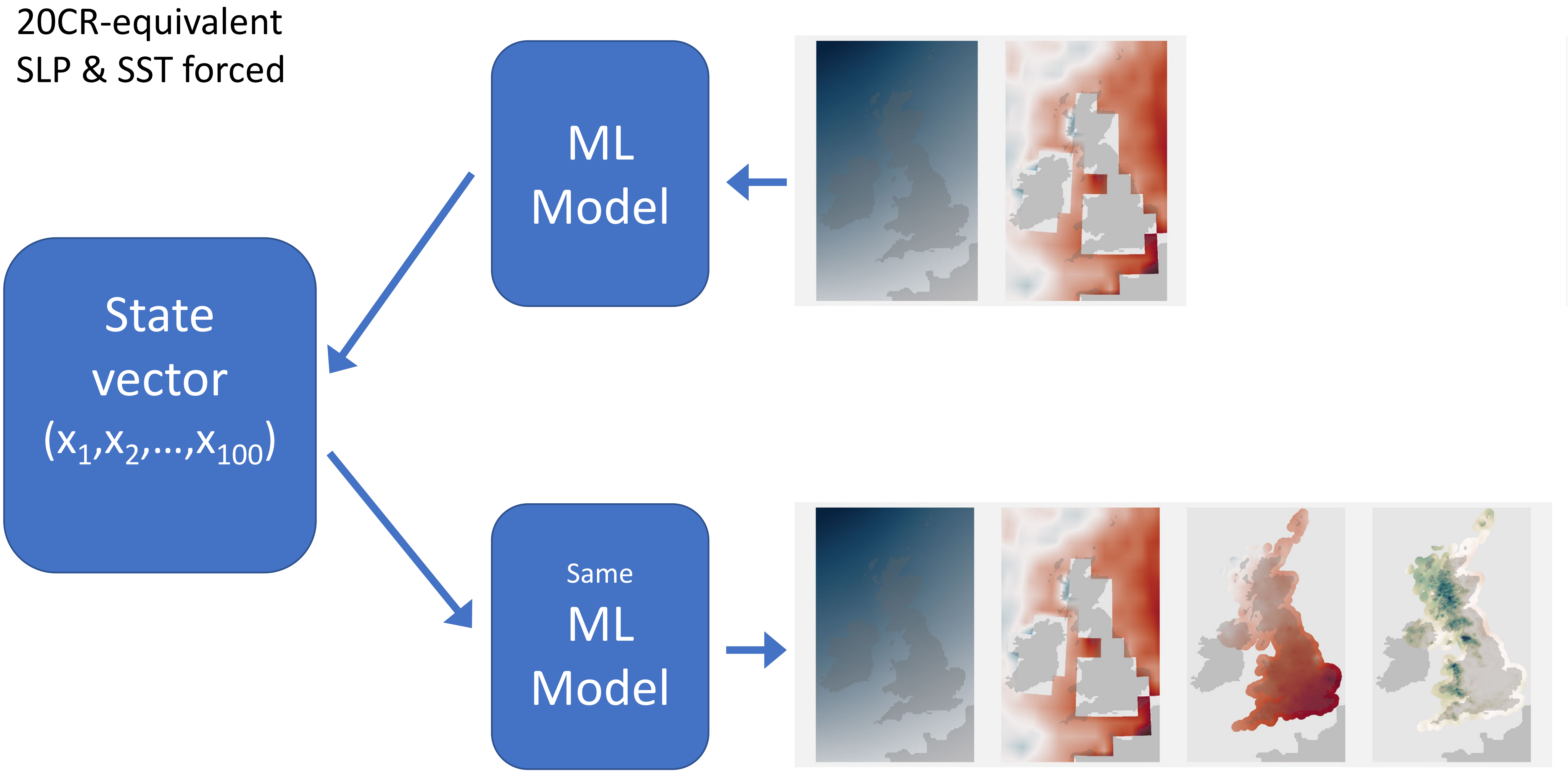
Middle: ML model output

Right: Target (x) v. model (y)





Results of assimilating SST only. Black – target, red – model. Means over the whole reconstructed field.

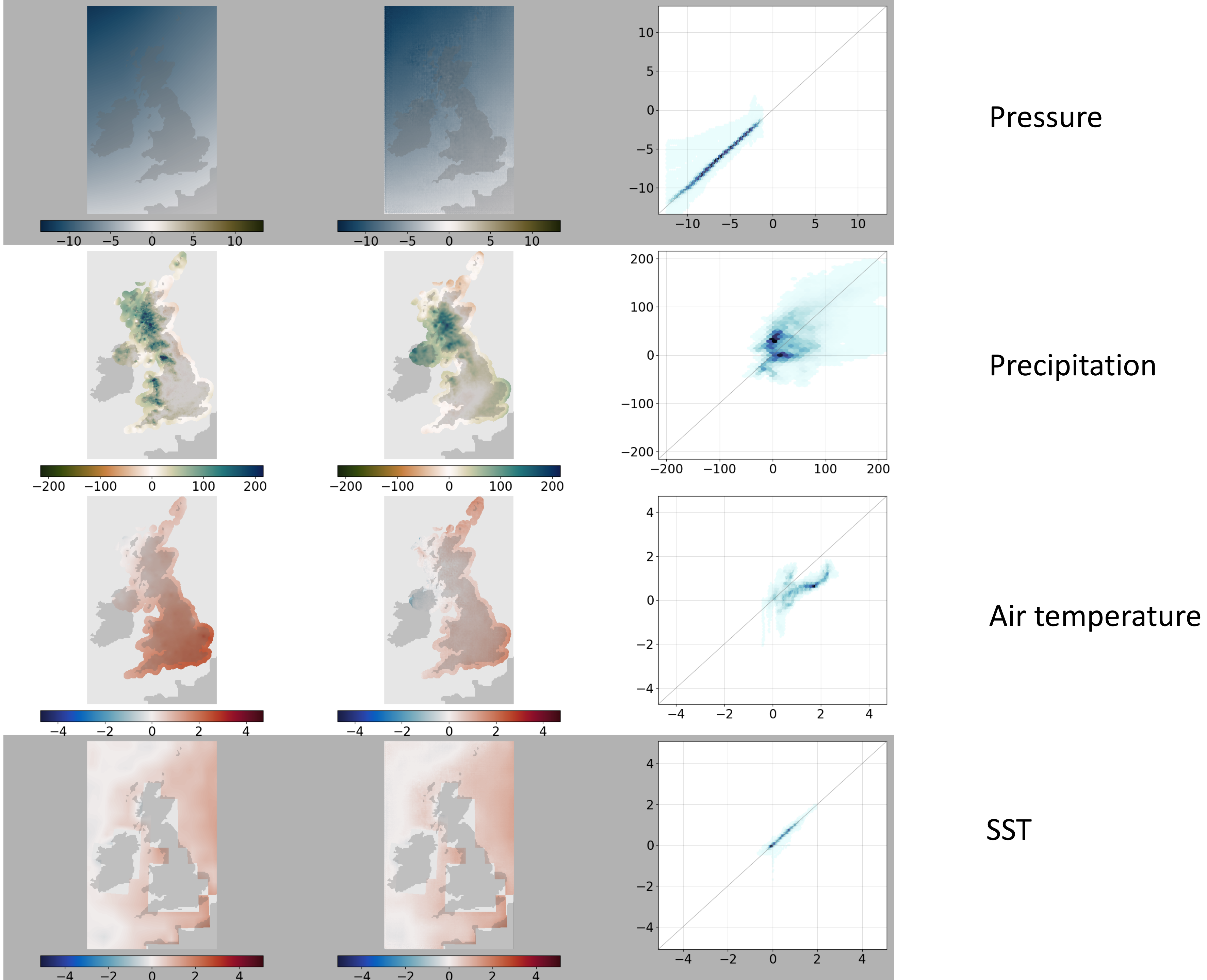

20CR-equivalent
SLP & SST forced
ML
Model
State
vector
$(x_1,x_2,\ldots,x_{100})$
Same
ML
Model

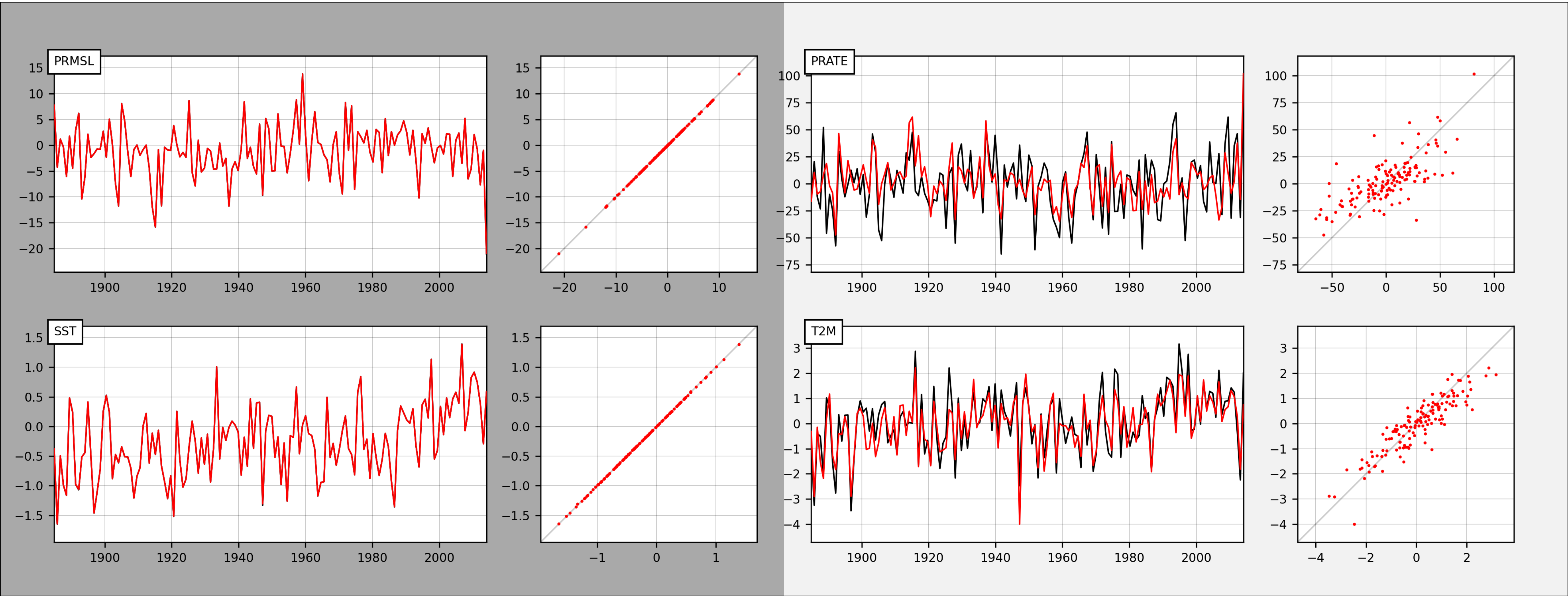
SST and PRMSL assimilation for one test month (1969-01).

Left: Target (truth)

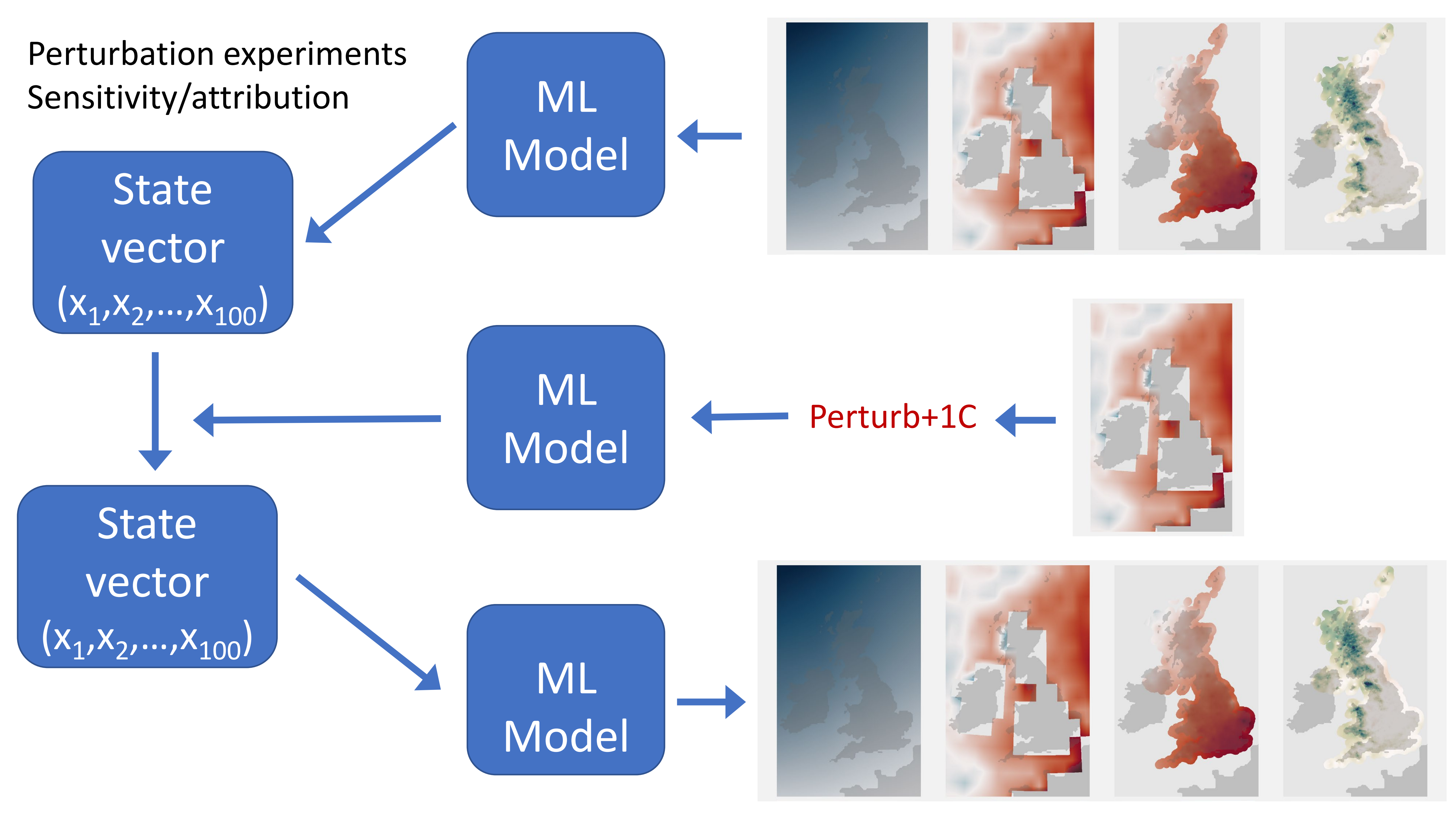
Middle: ML model output

Right: Target (x) v. model (y)





Results of assimilating SST and PRMSL. Black – target, red – model. Means over the whole reconstructed field.

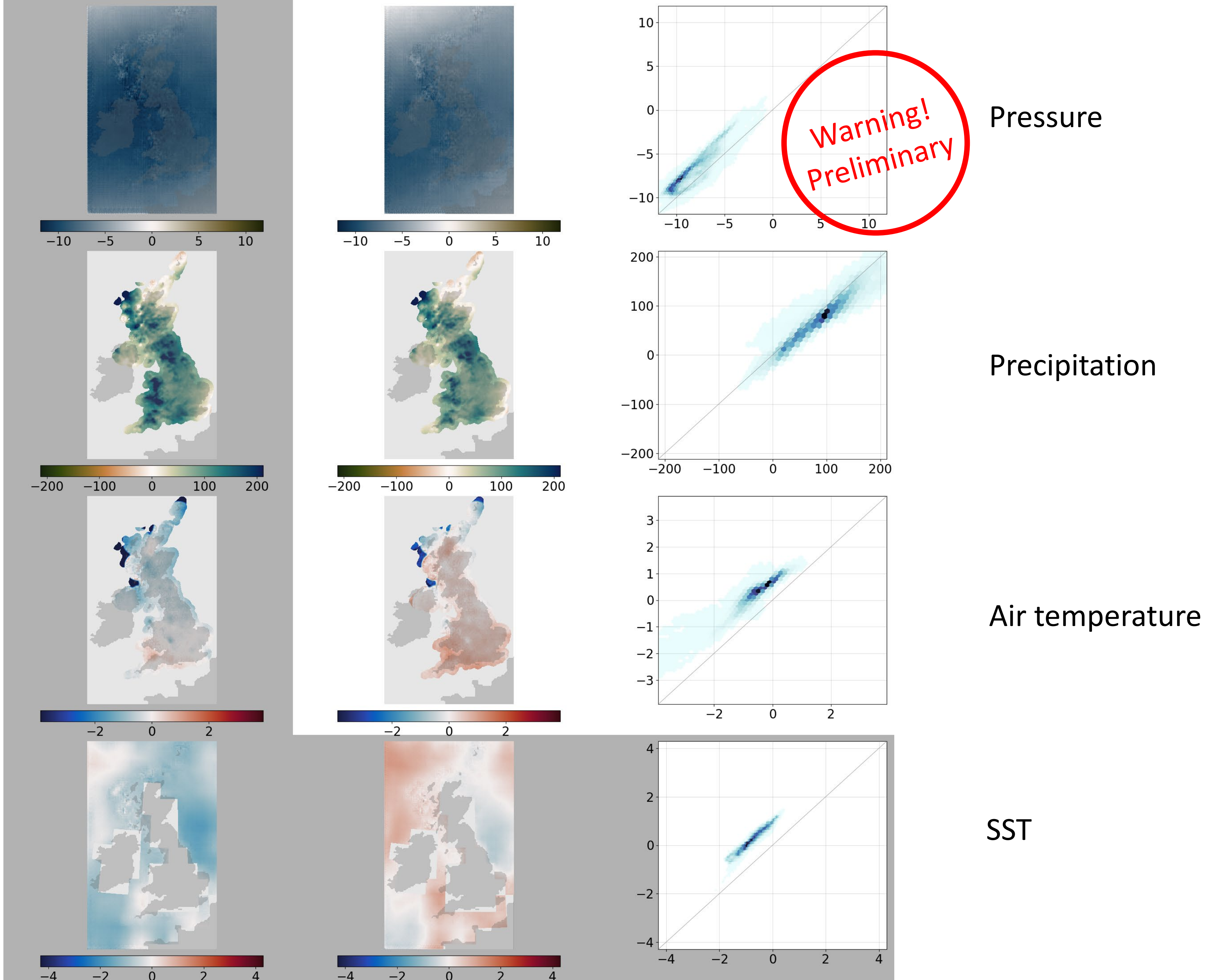

Perturbation experiments
Sensitivity/attribution
ML Model
State vector $(x_1, x_2, \ldots, x_{100})$
ML Model
Perturb+1C
State vector $(x_1, x_2, \ldots, x_{100})$
ML Model

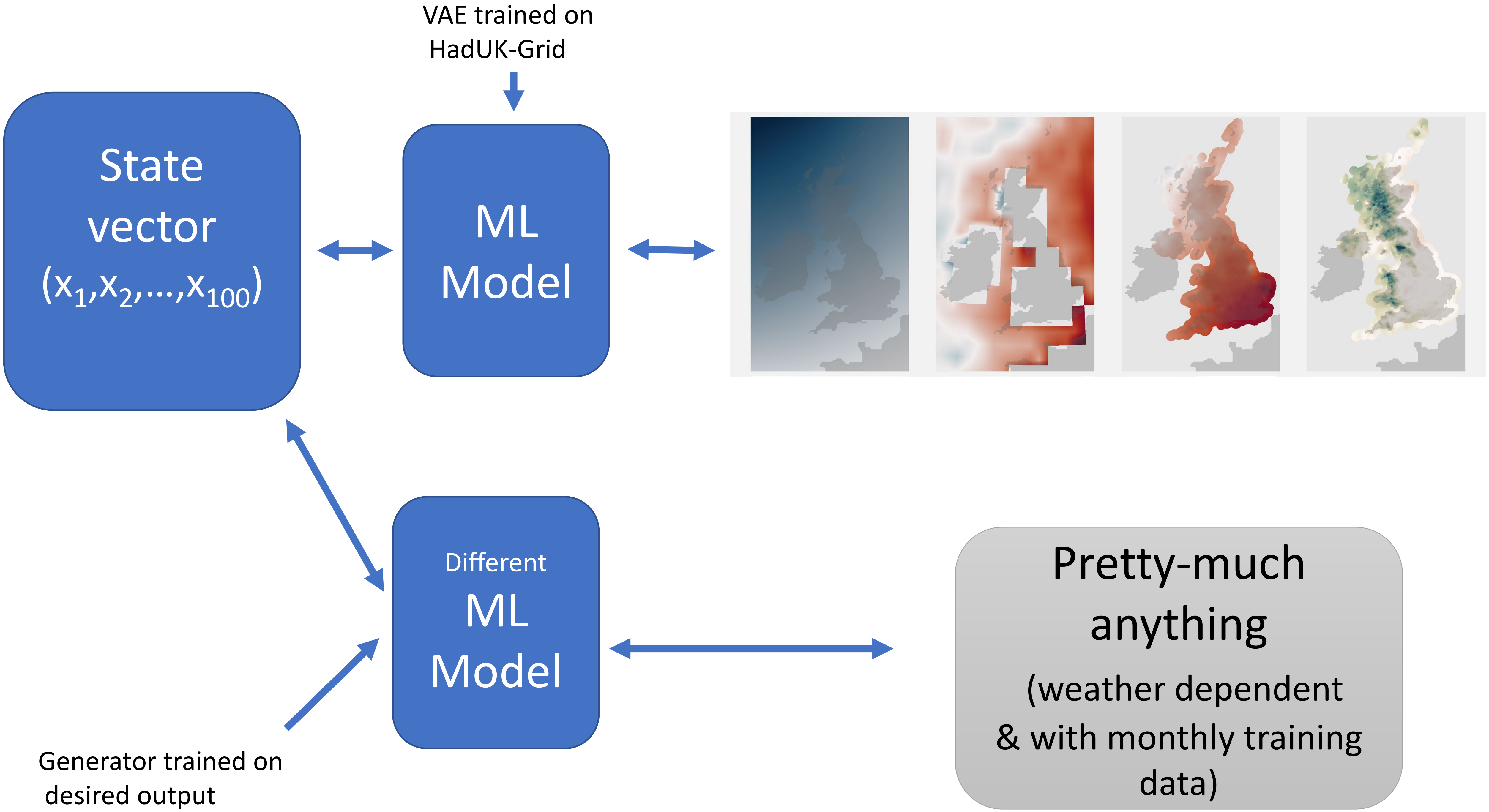
SST perturbation effect for one test month (1903-10: the wettest month on record).

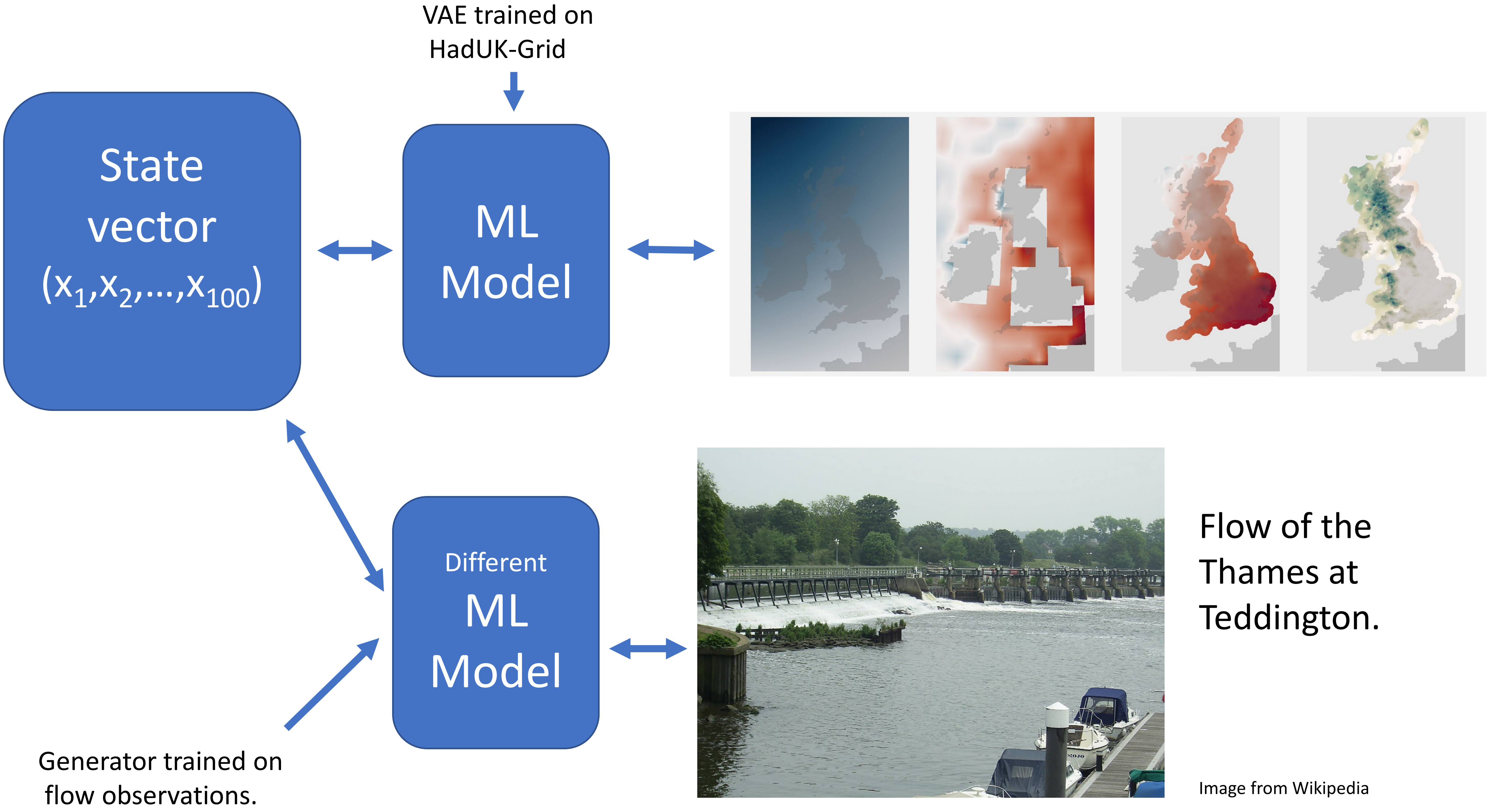
Left: ML model output after Assimilating all fields.

Middle: ML model output after Assimilating perturbed SST (+1C)

Right: Observed (x) v. perturbed (y)



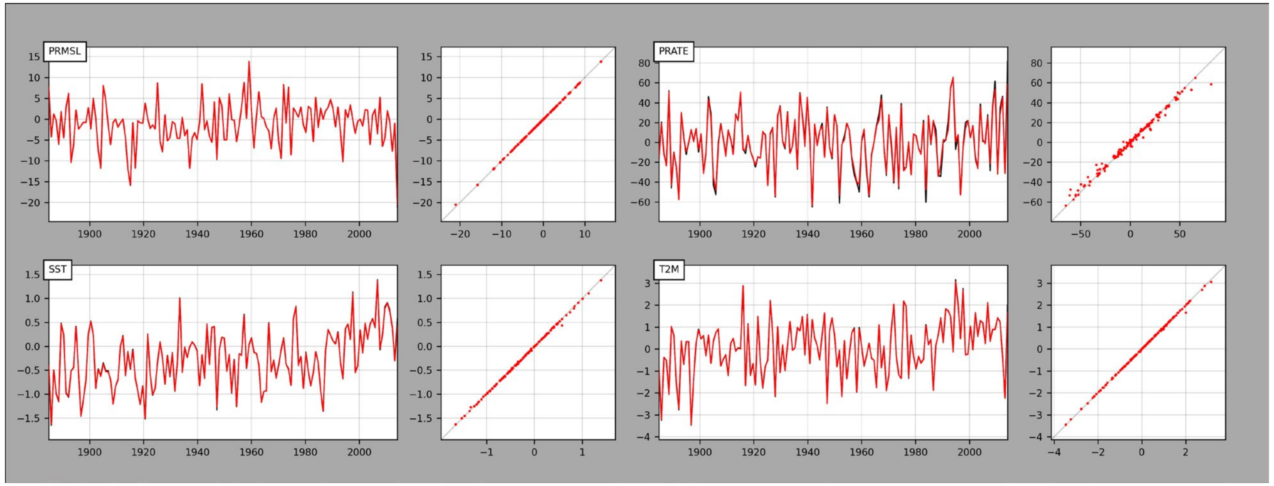

VAE trained on HadUK-Grid
State vector $(x_1,x_2,\ldots,x_{100})$
ML Model
Different ML Model
Generator trained on desired output
Pretty-much anything
(weather dependent & with monthly training data)


VAE trained on HadUK-Grid
State vector $(x_1, x_2, \ldots, x_{100})$
ML Model
Different ML Model
Generator trained on flow observations.
Flow of the Thames at Teddington.
Image from Wikipedia

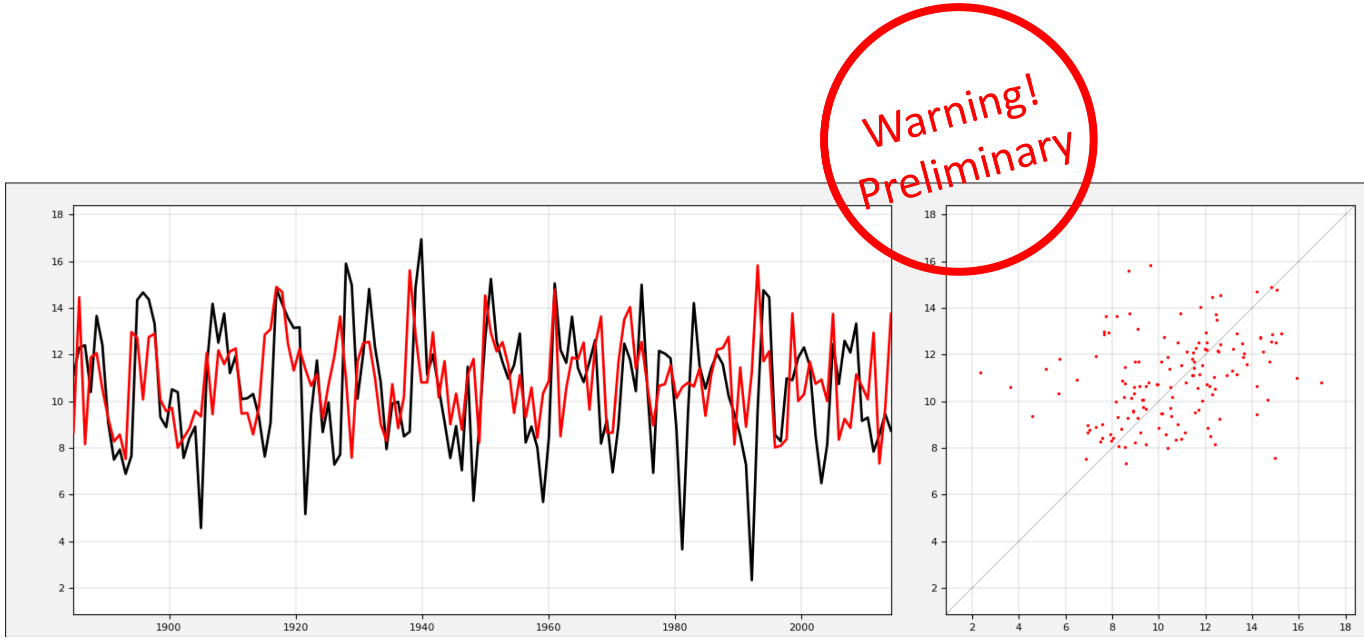
VAE trained on HadUK-Grid

State vector $(x_1, x_2, \ldots, x_{100})$

ML Model



Different ML Model



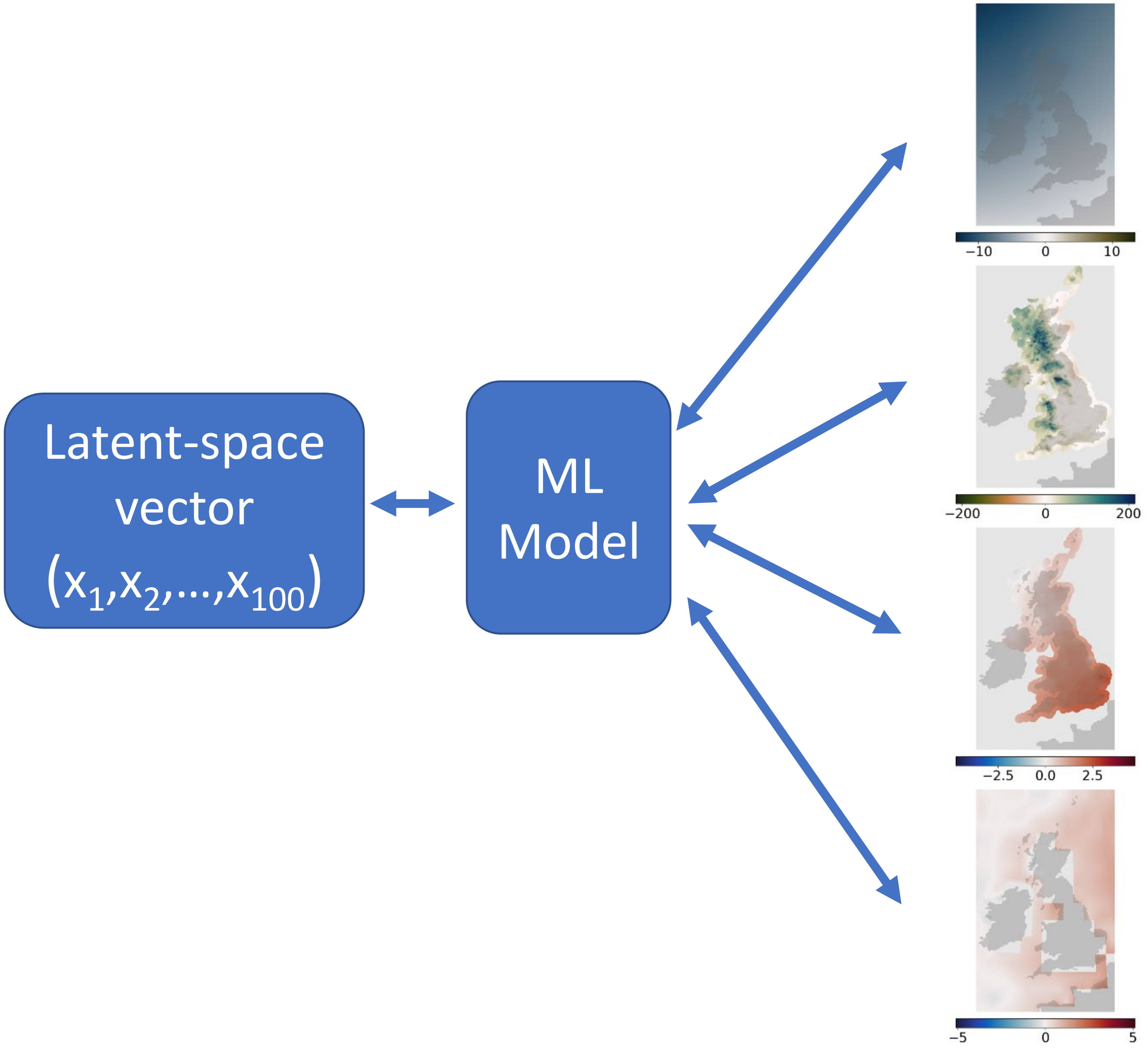
2-layer perceptron trained on
Flow data (Thames at Kingston) from NRFA:
https://nrfa.ceh.ac.uk/data/station/info/39001

Trained ML model to represent multivariate climate state as a 100-dimensional latent-space (LS) vector. ML model is a Deep Convolutional Variational AutoEncoder, trained on HadUK-Grid.

ML model is bidirectional – can estimate LS vector for a month from an arbitrary subset of real climate state, and then recover full climate state from LS vector. => Data Assimilation: recover full state from sparse observations.

ML model is ~1,000,000 times as fast as an equivalent GCM => many applications in reanalysis and climate modelling.

Straightforwardly extensible to add other weather fields or arbitrary impacts variables => Climate Services.



This work was supported by the Met Office Hadley Centre Climate Programme funded by BEIS, and by the UK-China Research & Innovation Partnership Fund through the Met Office Climate Science for Service Partnership (CSSP) China as part of the Newton Fund.

Training data used came from the HadUK-Grid & 20CR.

The software produced was built on the TensorFlow platform

The models were trained on the Isambard UK National Tier-2 HPC Service operated by GW4 and the UK Met Office, and funded by EPSRC (EP/P020224/1).

philip.brohan@metoffice.gov.uk

@philipbrohan