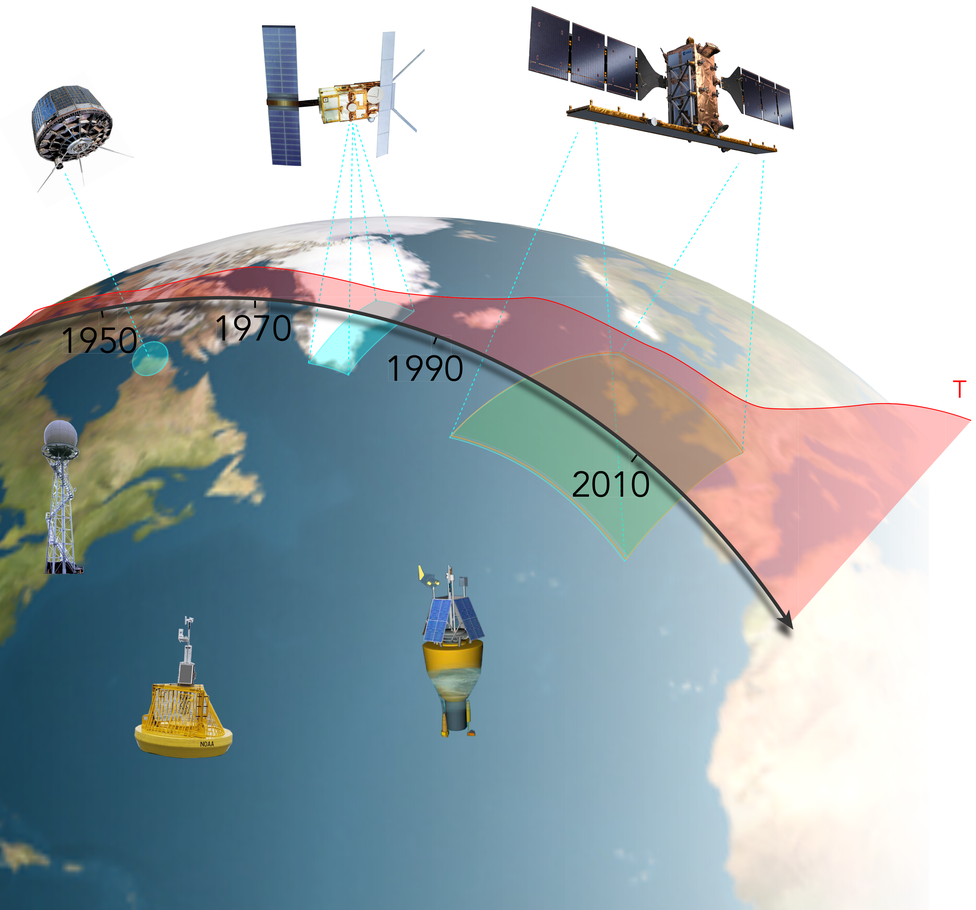
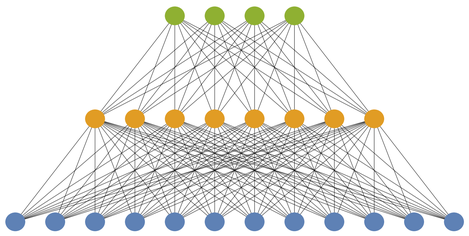


AtmoRep

Large scale representation learning of atmospheric dynamics

Christian Lessig, Ilaria Luise, Martin Schultz, et al.



applications

large scale representation learning

scientific insight

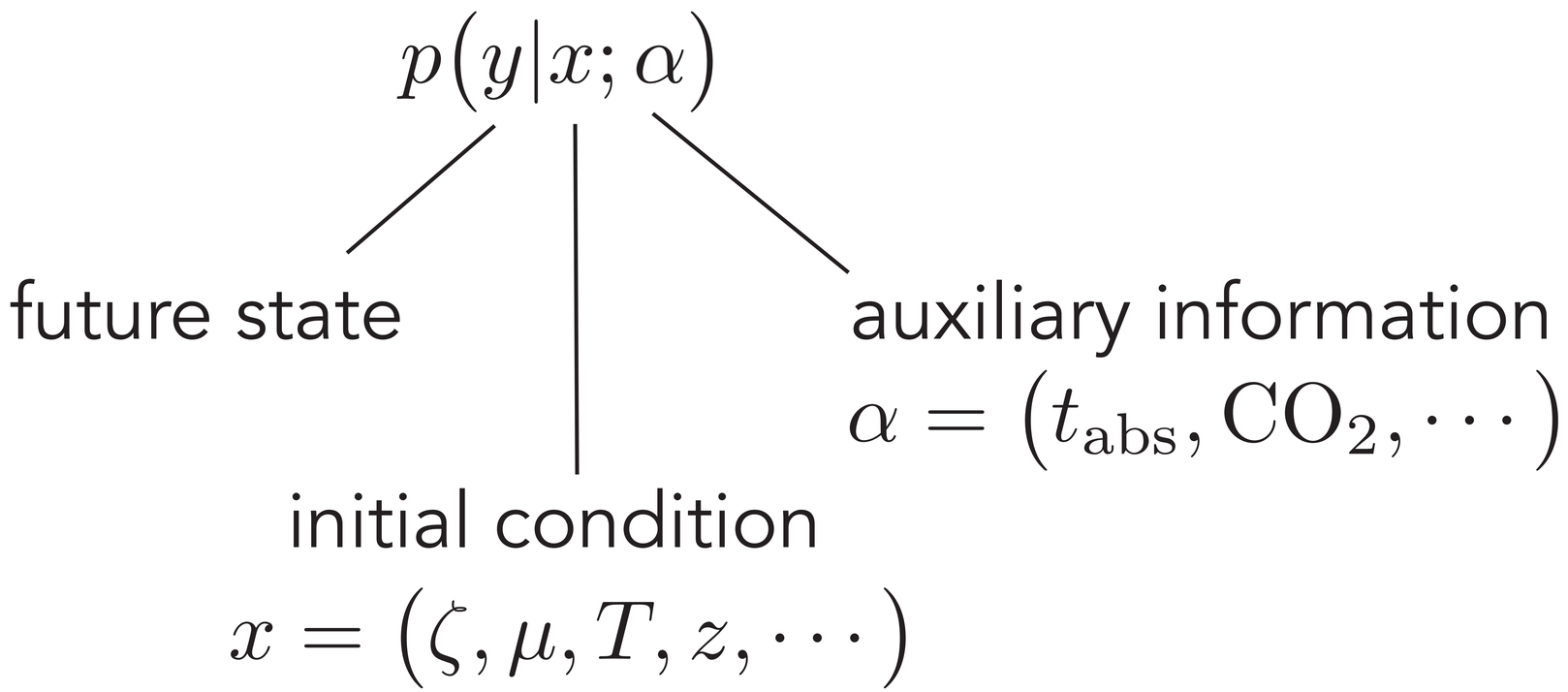
Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

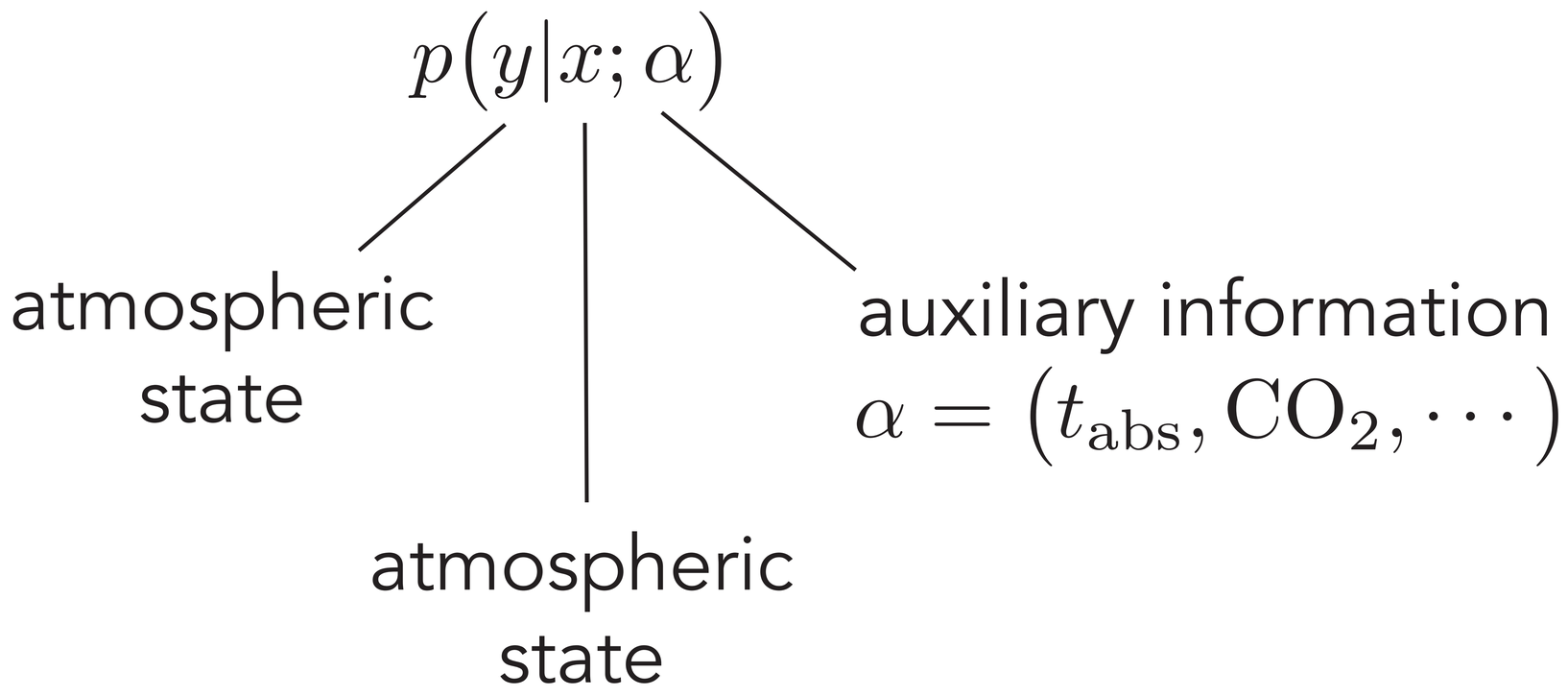
Theoretical formulation

- Atmosphere as stochastic dynamical system:



Theoretical formulation

- Atmosphere as stochastic dynamical system:



Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

  - Forecasting, downscaling, interpolation: $p\big(y|x;\alpha\big)$

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

  - Forecasting, downscaling, interpolation: $p\big(y|x;\alpha\big)$
  - Counterfactuals: $p\big(y|x;\hat{\alpha}\big)$

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

  - Forecasting, downscaling, interpolation: $p\big(y|x;\alpha\big)$
  - Counterfactuals: $p\big(y|x;\hat{\alpha}\big)$
  - Climate: $p_\alpha(y) = \int p\big(y|x;\alpha\big)\, p(x)\, \mathrm{d}x$

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

  - Medium/extended range forecasting:

$$p(x_T|x_0;\alpha) = \prod_{i=1}^{T} p_\theta(x_{i+1}|x_i;\alpha)$$

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p(y|x;\alpha)$$

highly complex, in-stationary distribution

  - Medium/extended range forecasting:

$$p(x_T|x_0;\alpha) = \prod_{i=1}^{T} p_\theta(x_{i+1}|x_i;\alpha)$$

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

- Numerical statistical atmospheric model:

$$\tilde{p}\big(\tilde{y}|\tilde{x};\tilde{\alpha}\big)$$

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

- Numerical statistical atmospheric model:

$$\tilde{p}\big(\tilde{y}|\tilde{x};\tilde{\alpha}\big) \approx p\big(y|x;\alpha\big)$$

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

- Numerical statistical atmospheric model:

$$\tilde{p}\big(\tilde{y}|\tilde{x};\tilde{\alpha}\big) \approx p\big(y|x;\alpha\big)$$

approx. future state approx. initial condition

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

- Numerical statistical atmospheric model:

$$\tilde{p}\big(\tilde{y}|\tilde{x};\tilde{\alpha}\big) \approx p\big(y|x;\alpha\big)$$

highly complex, instationary distribution

Theoretical formulation

- Atmosphere as stochastic dynamical system:

$$p\big(y|x;\alpha\big)$$

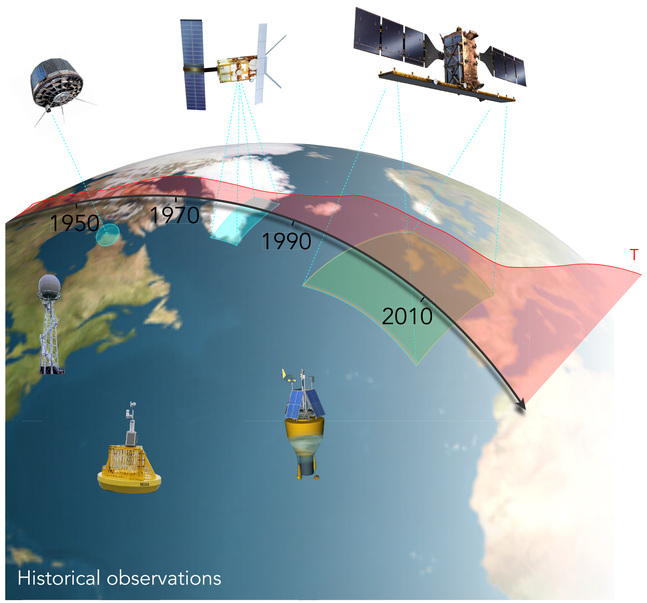
- Numerical statistical atmospheric model:

very large neural network

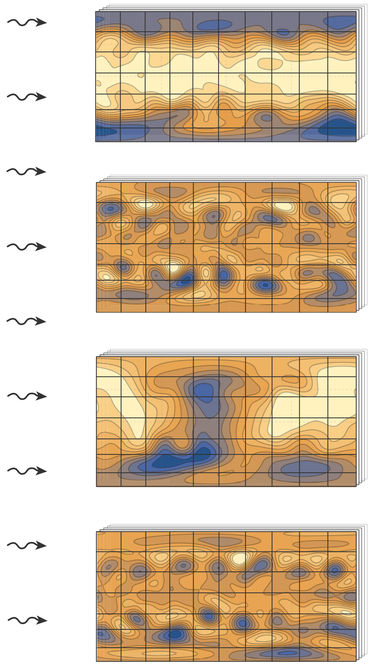
$$\tilde{p}_\theta\big(\tilde{y}|\tilde{x};\tilde{\alpha}\big) \approx p\big(y|x;\alpha\big)$$

highly complex, instationary distribution

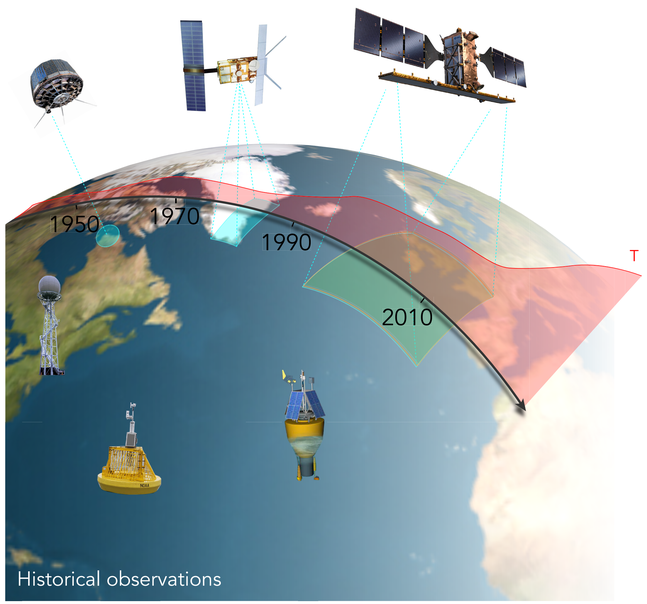
AtmoRep



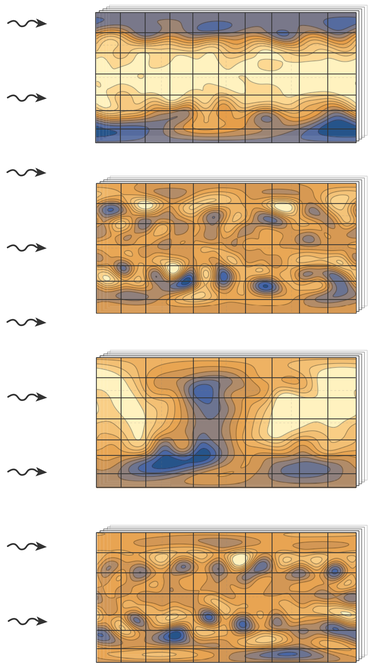
ERA5 reanalysis



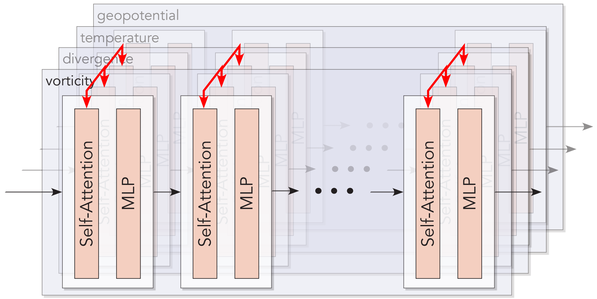
AtmoRep



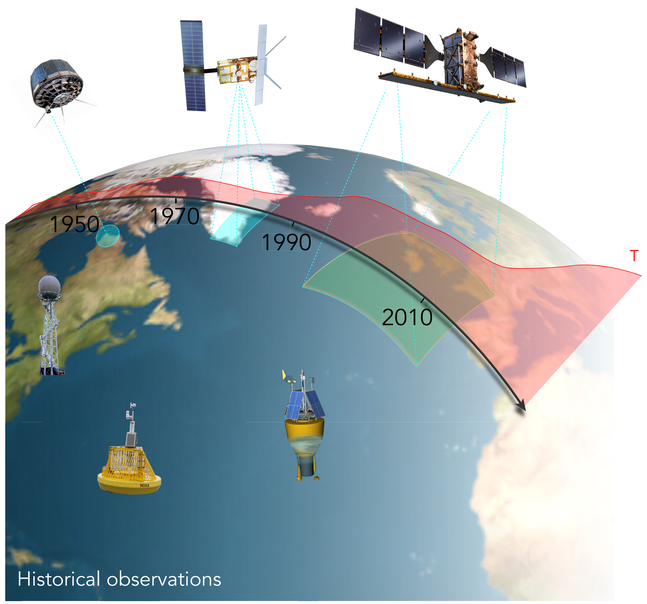
ERA5 reanalysis



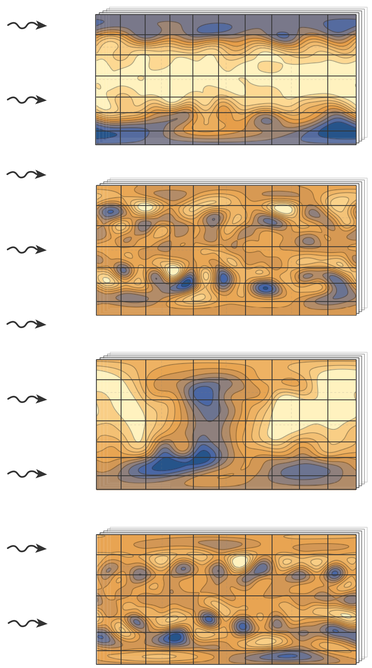
$$p_\theta(y|x)$$



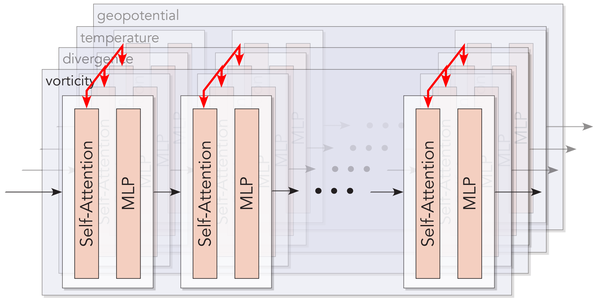
AtmoRep



ERA5 reanalysis

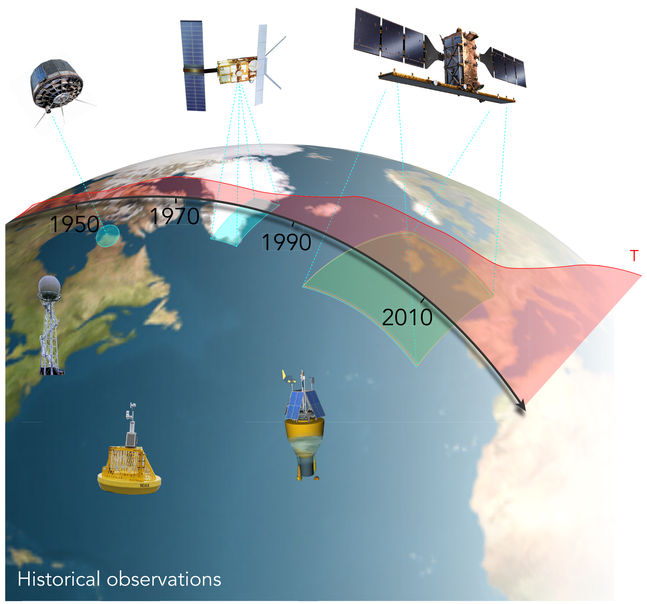


$$p_\theta(y|x)$$

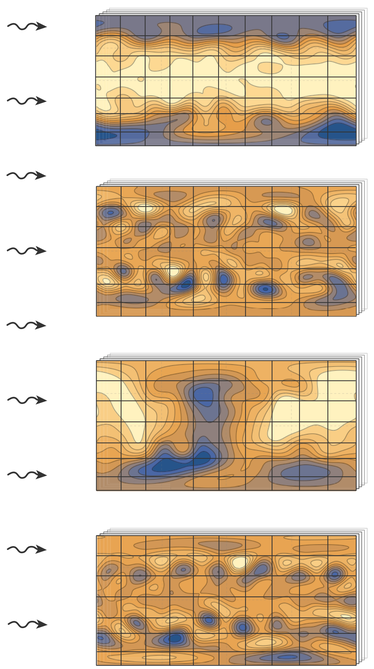


large transformer
with 3.5 x 10^9 parameters

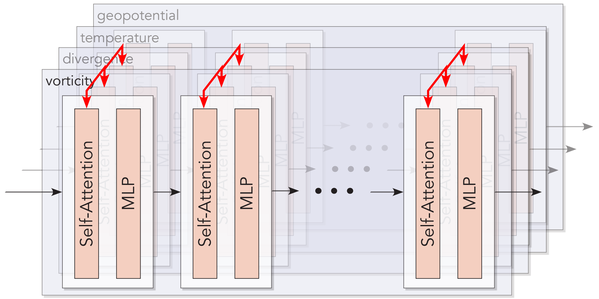
AtmoRep



ERA5 reanalysis

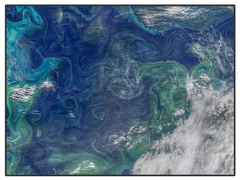


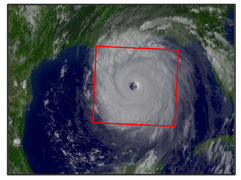
$p_\theta(y|x)$

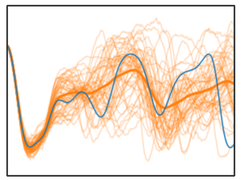


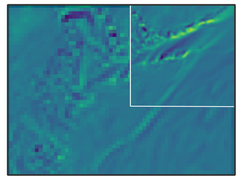
large transformer
with 3.5 x 10^9 parameters

applications

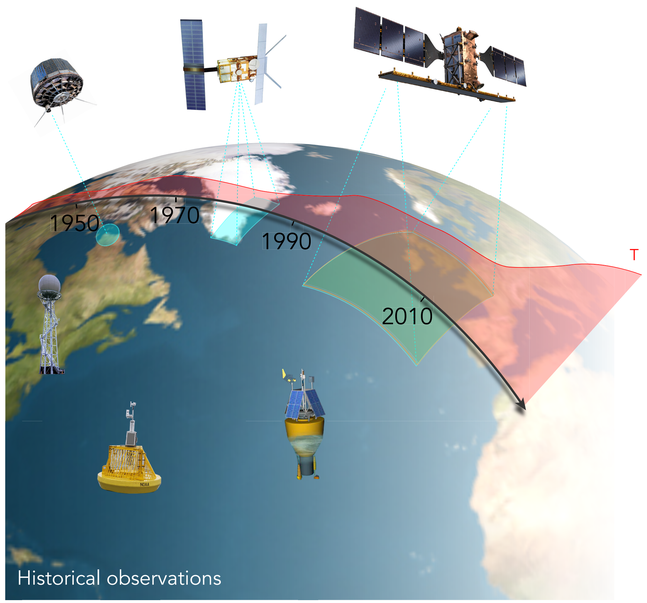

Forecasting


Impact analysis

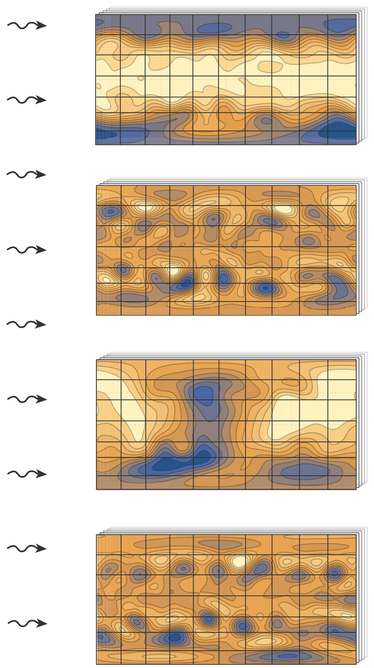

Statistical analysis


Downscaling

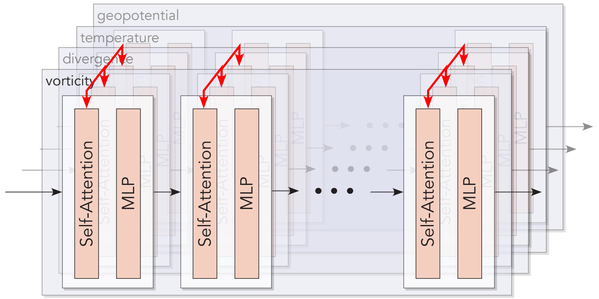
AtmoRep



ERA5 reanalysis

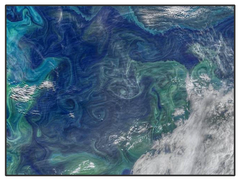


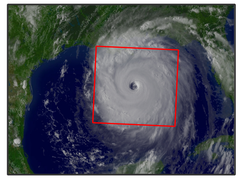
$p_\theta(y|x)$

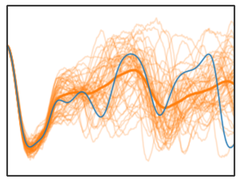


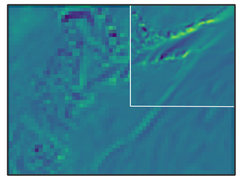
large transformer
with 3.5 x 10^9 parameters

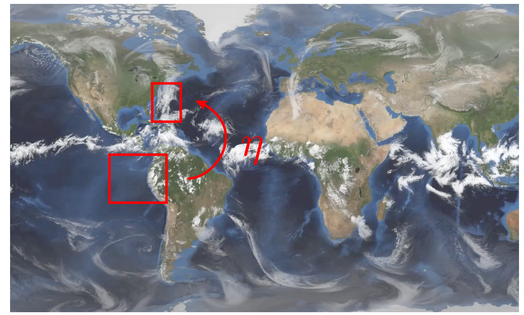
applications


Forecasting


Impact analysis


Statistical analysis

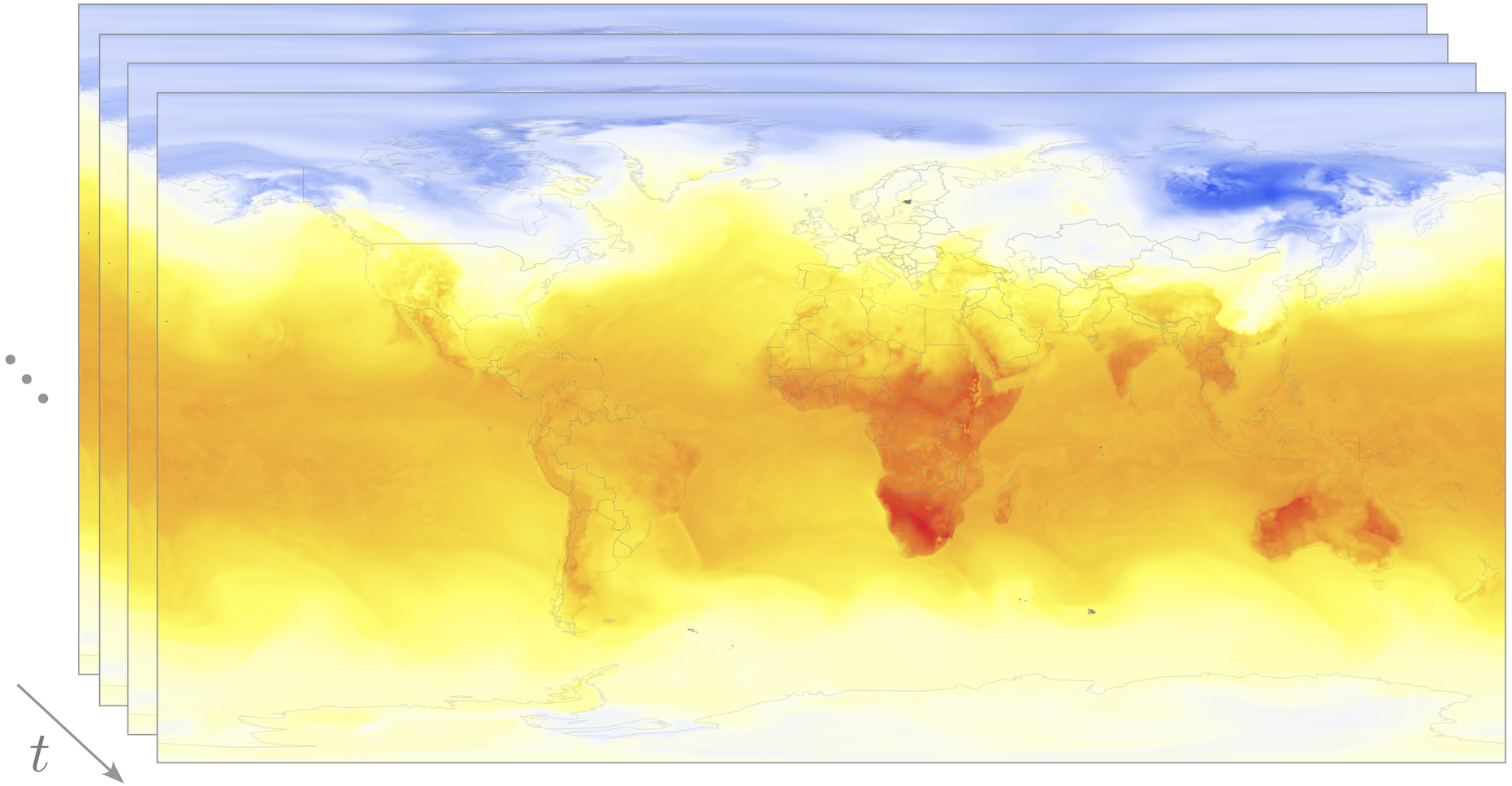

Downscaling



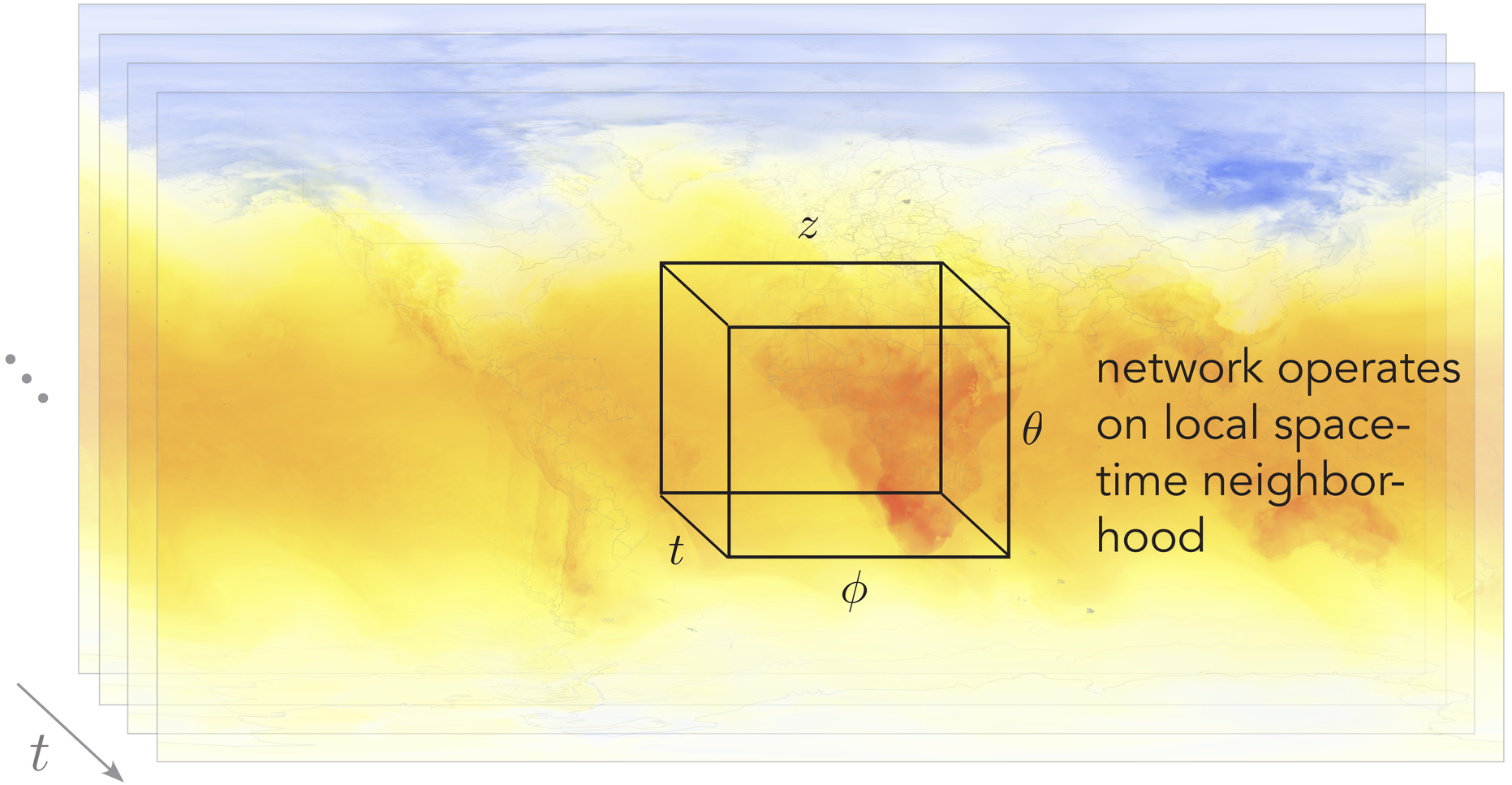
scientific insight

Network architecture

AtmoRep network architecture



AtmoRep network architecture



network operates on local space-time neighbor-hood

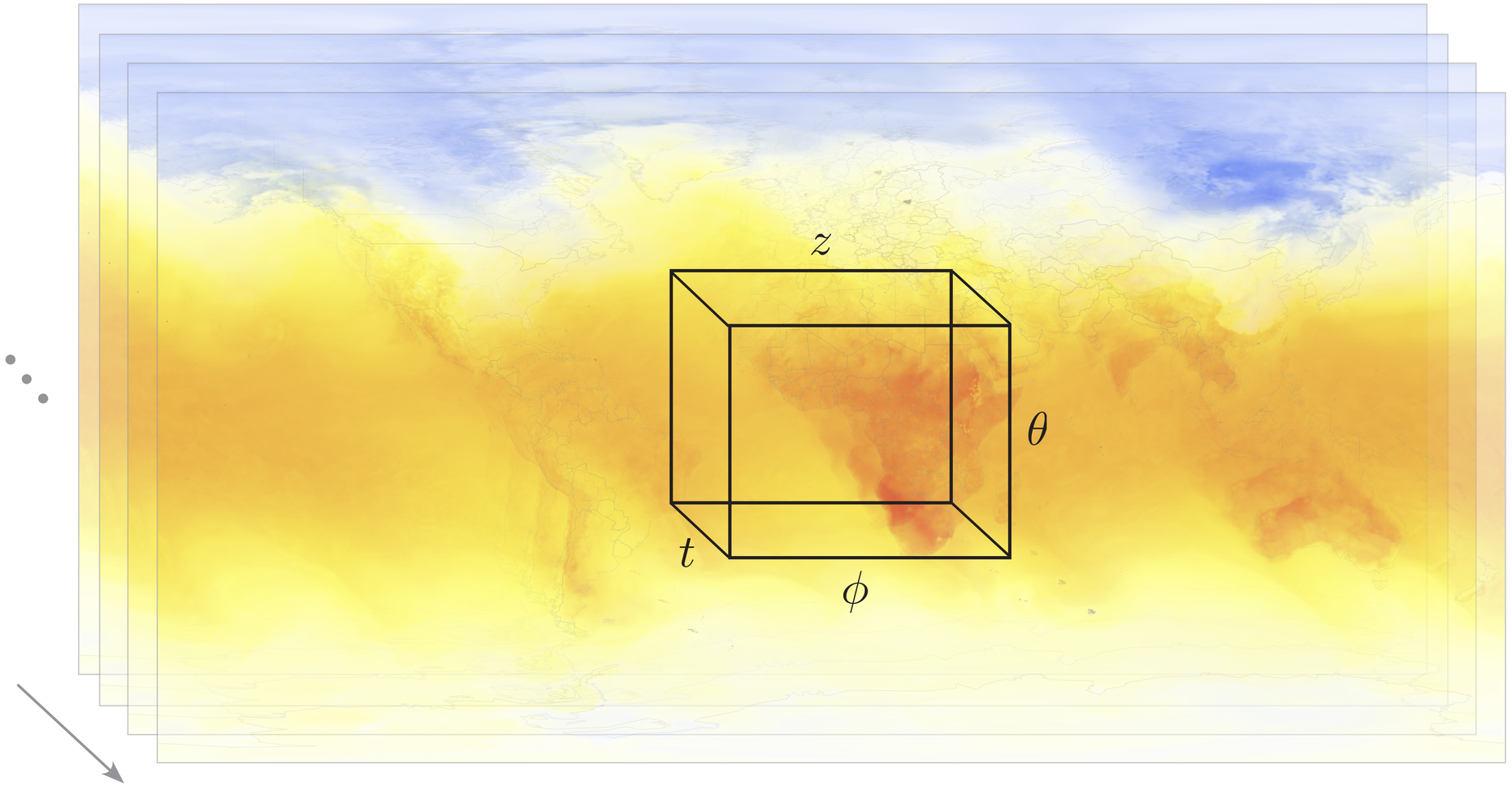
AtmoRep network architecture

- Network is local in space-time
  - Physics are universally valid
  - Machine learning model can be leaner and learn faster
  - Local particularities can be learned by providing time + space position as network input

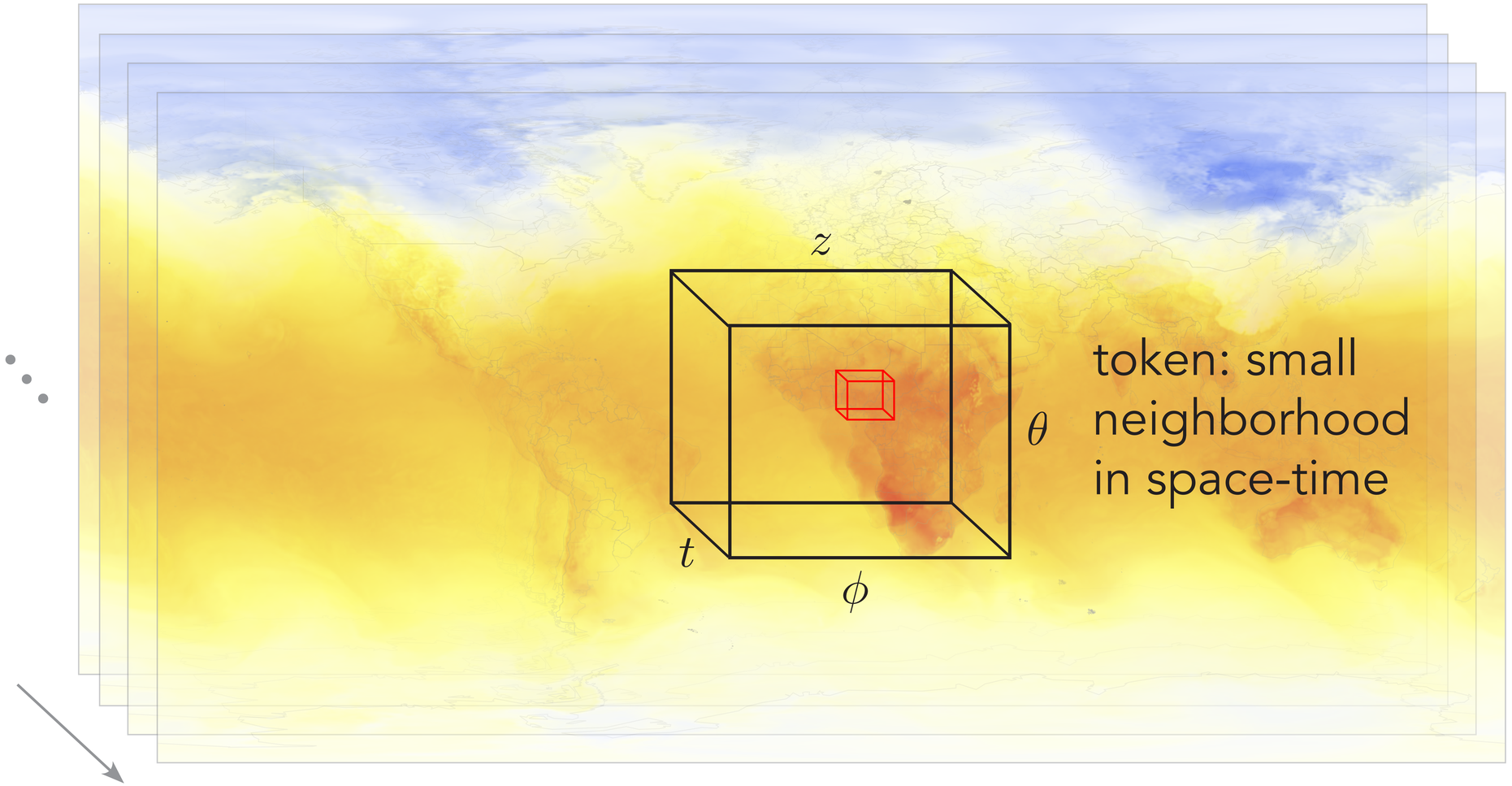
AtmoRep network architecture

- Network is local in space-time
  - Physics are universally valid
  - Machine learning model can be leaner and learn faster
  - Local particularities can be learned by providing time + space position as network input
- Network is an extended transformer architecture

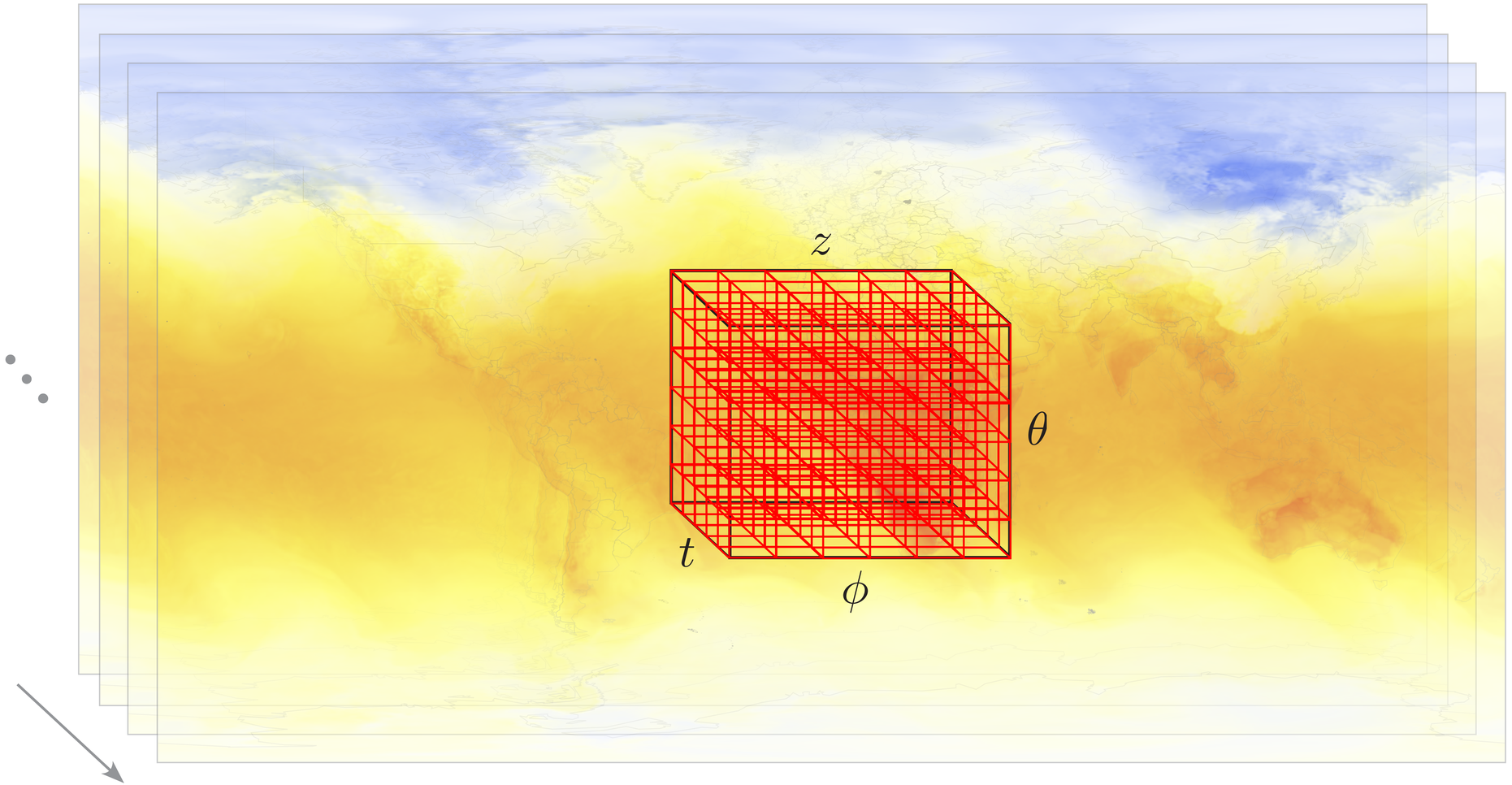
What is a token?



What is a token?

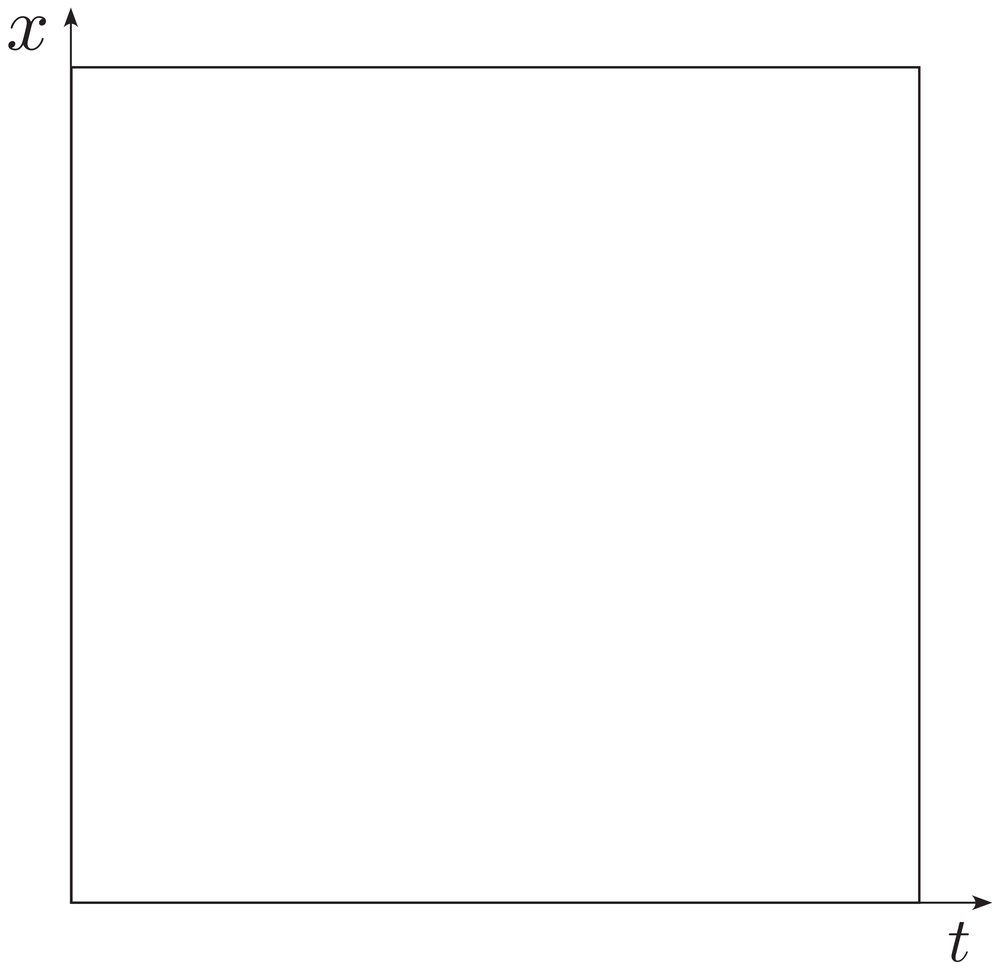


What is a token?



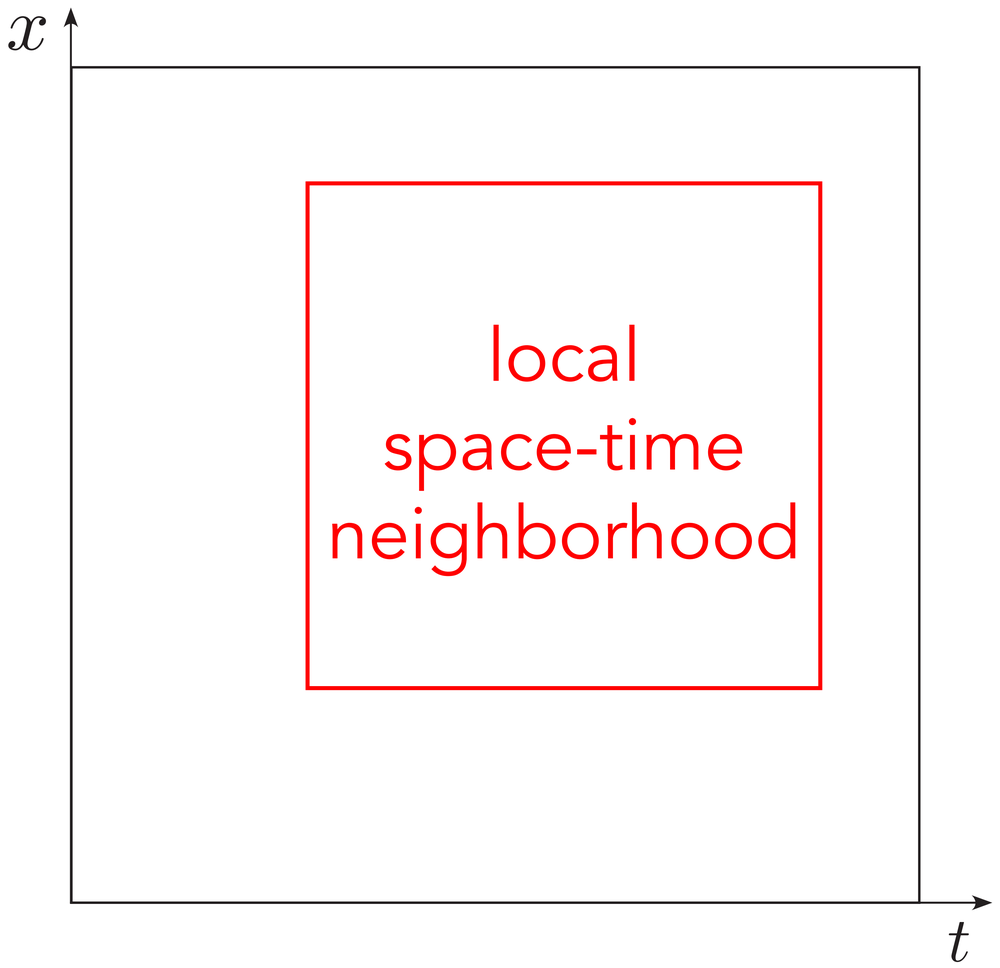
What is a token?

- Flatland view:



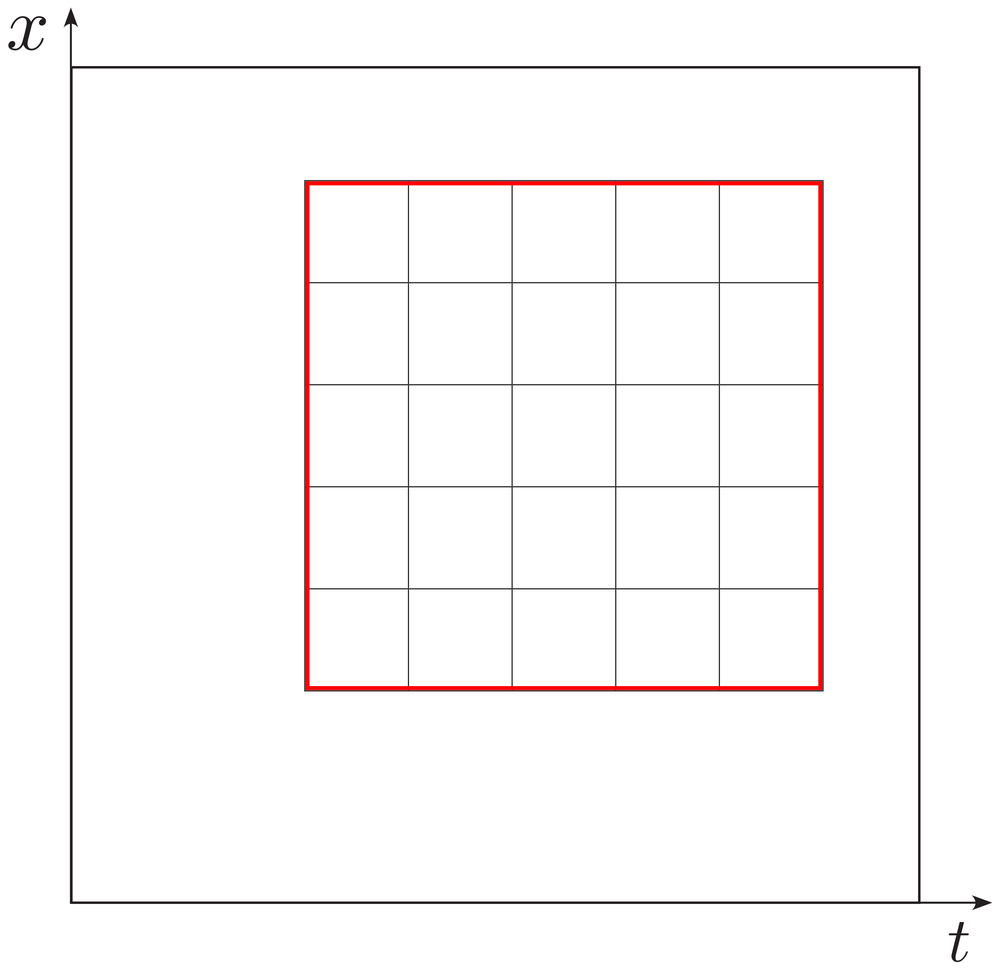
What is a token?

- Flatland view:



What is a token?

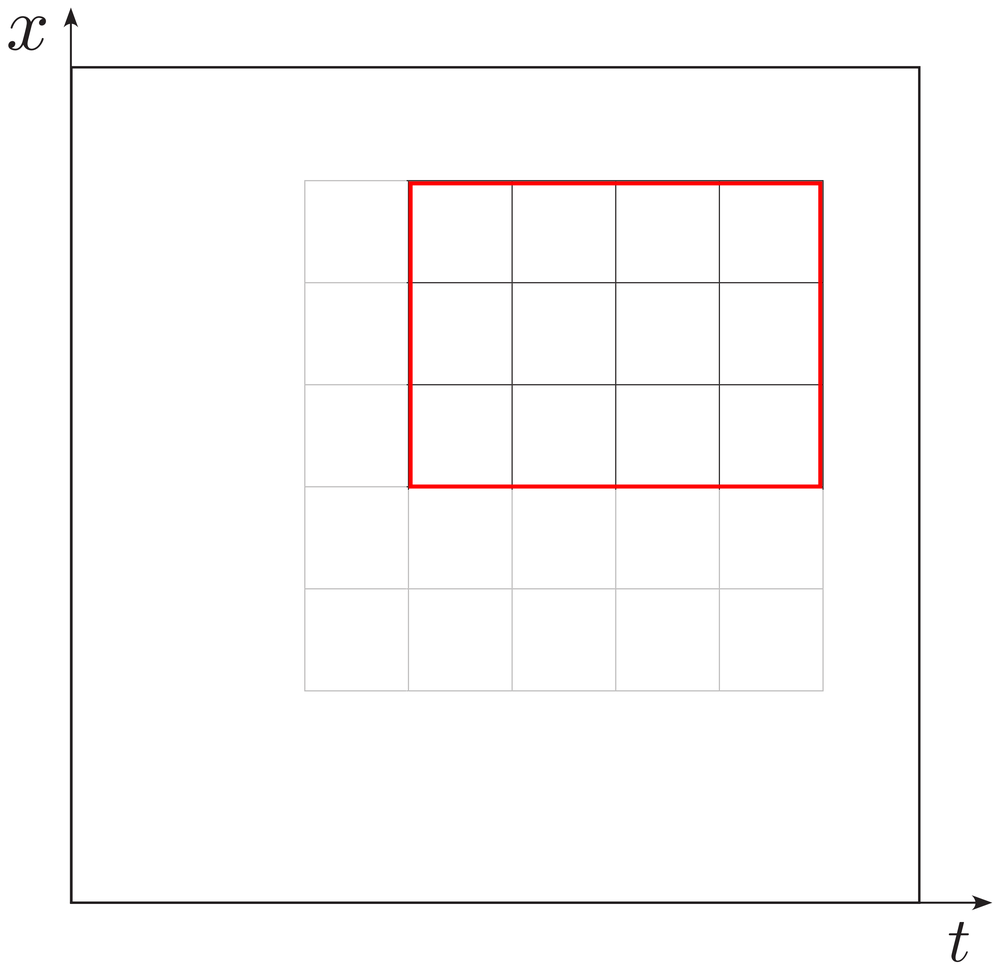
- Flatland view:



What is a token?

- Flatland view:

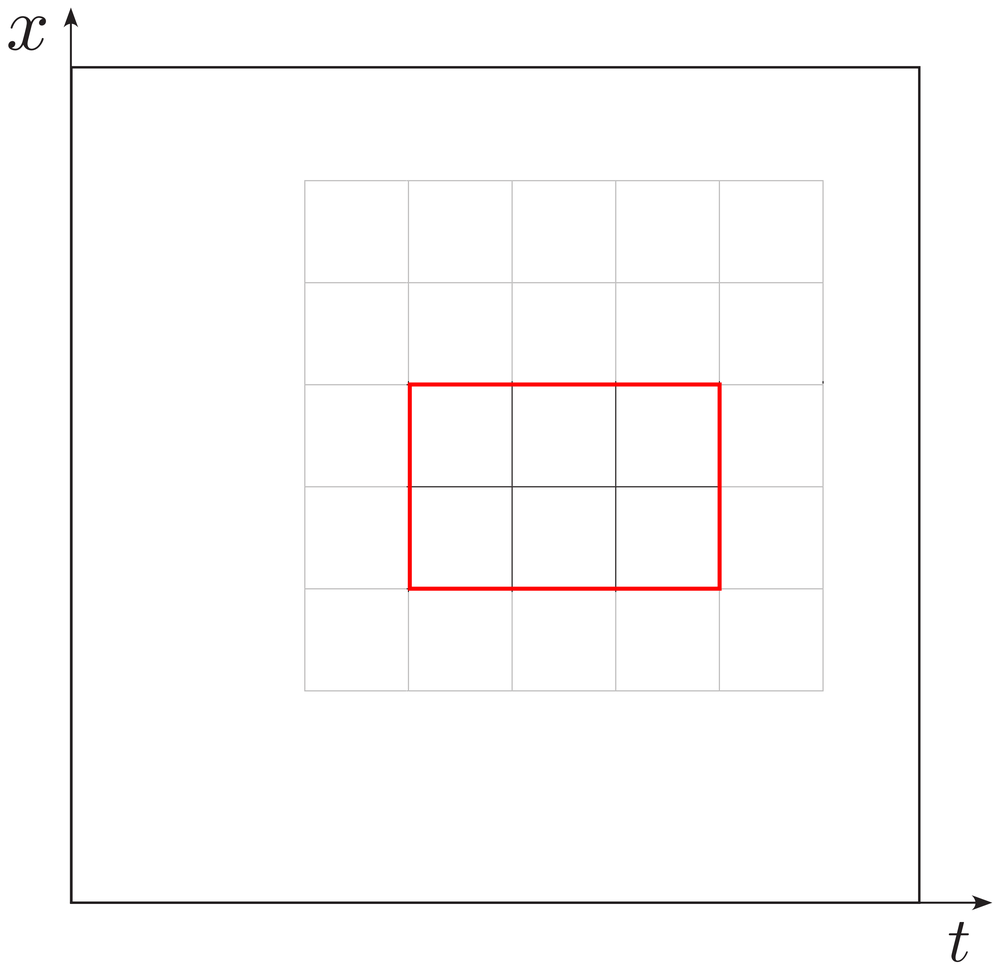
local window
size is flexible



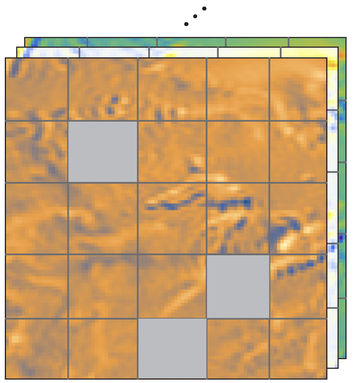
What is a token?

- Flatland view:

local window
size is flexible

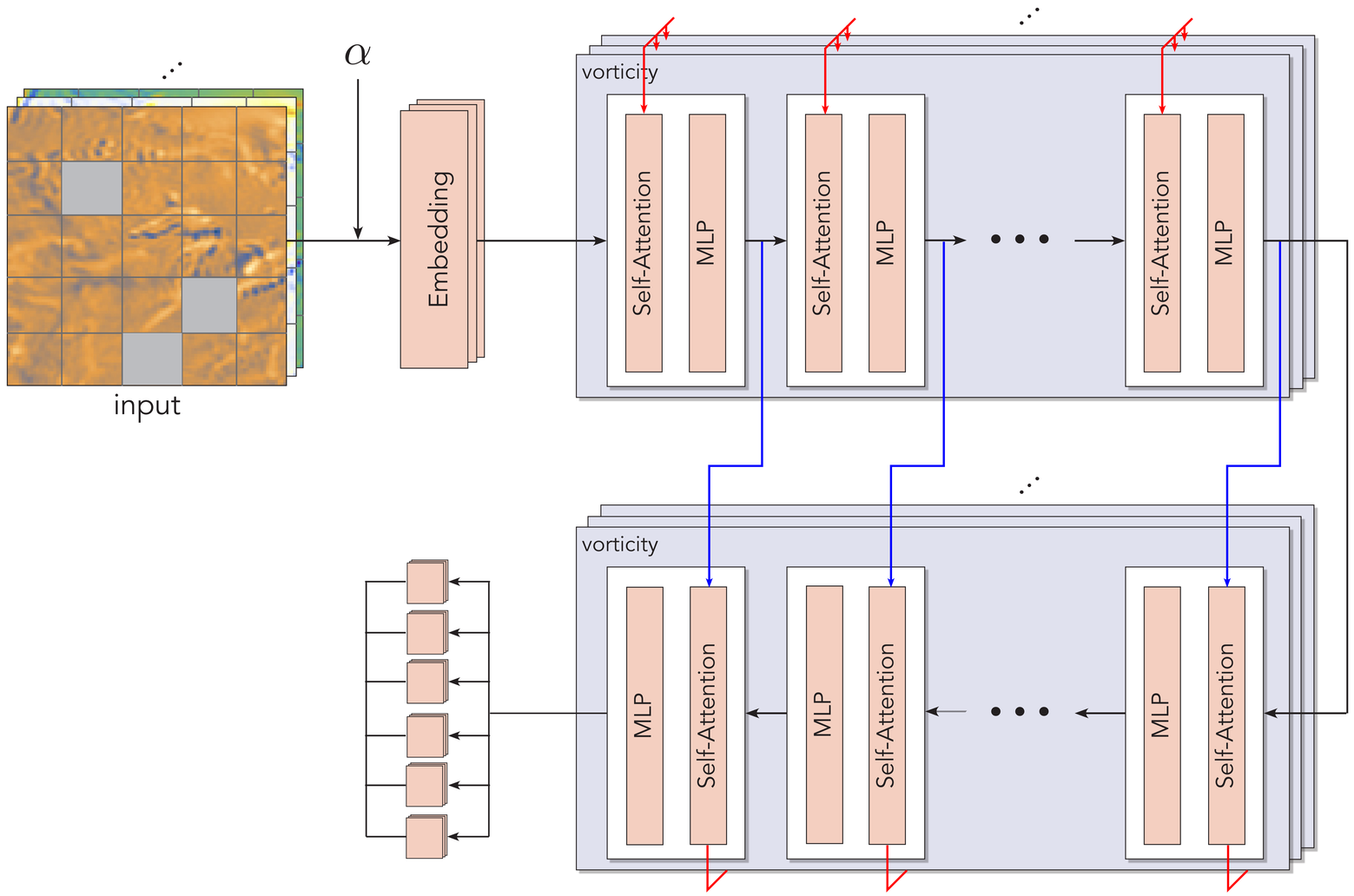


Multiformer

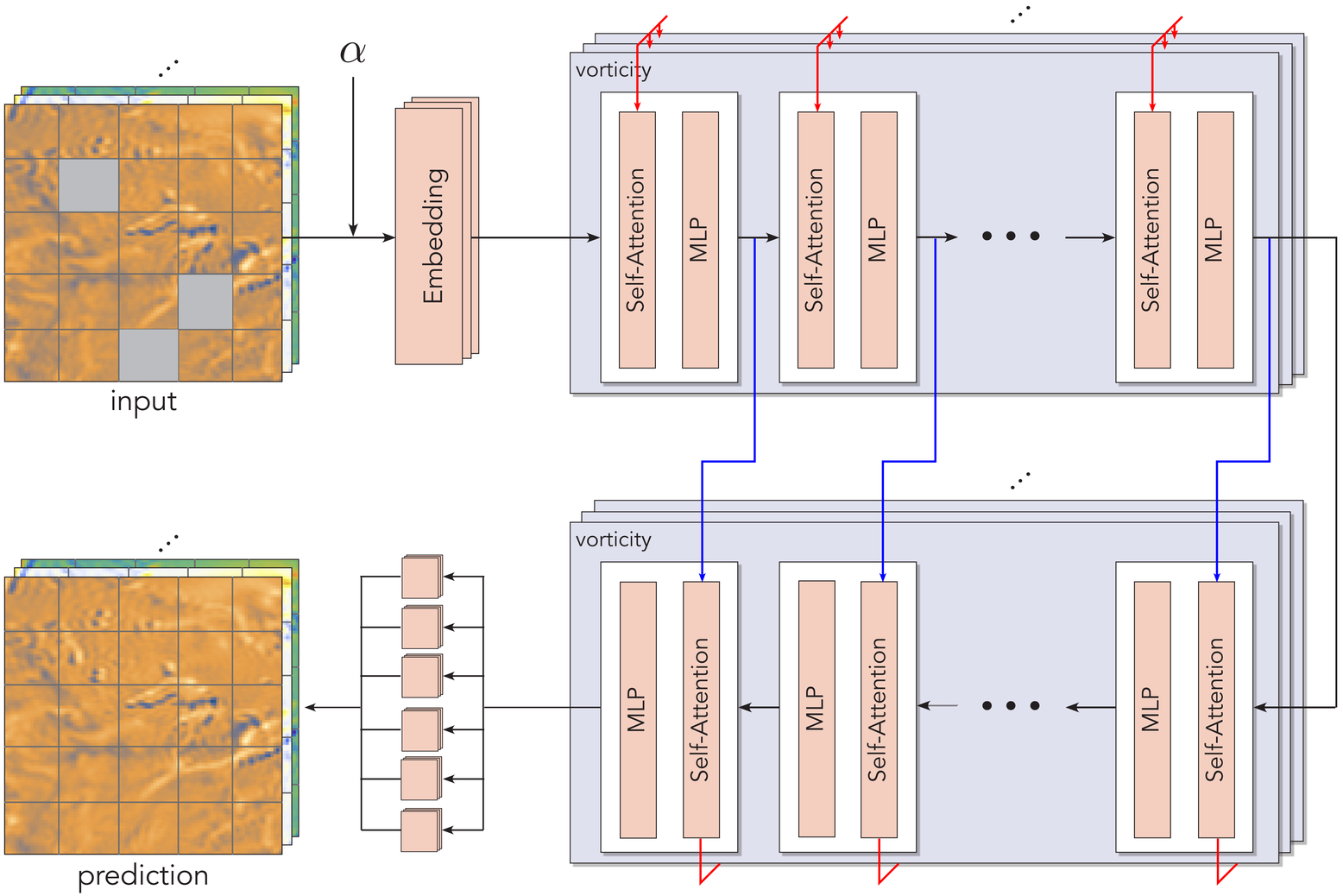


input

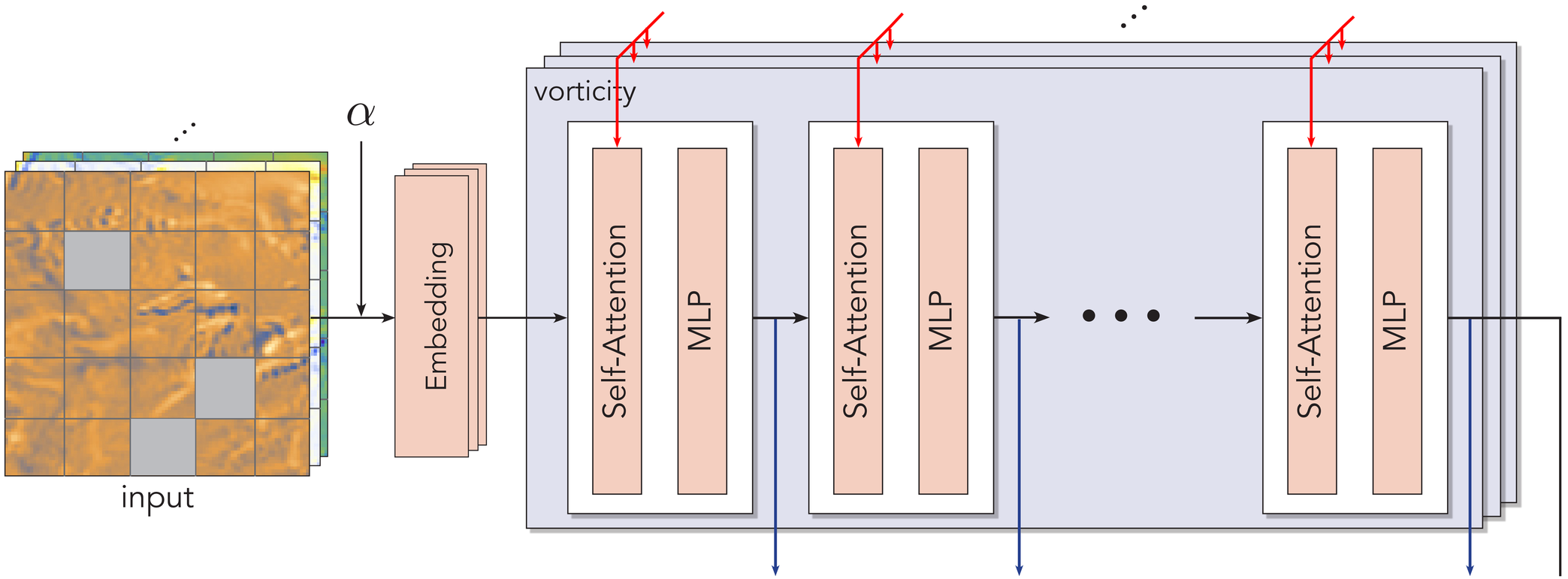
Multiformer



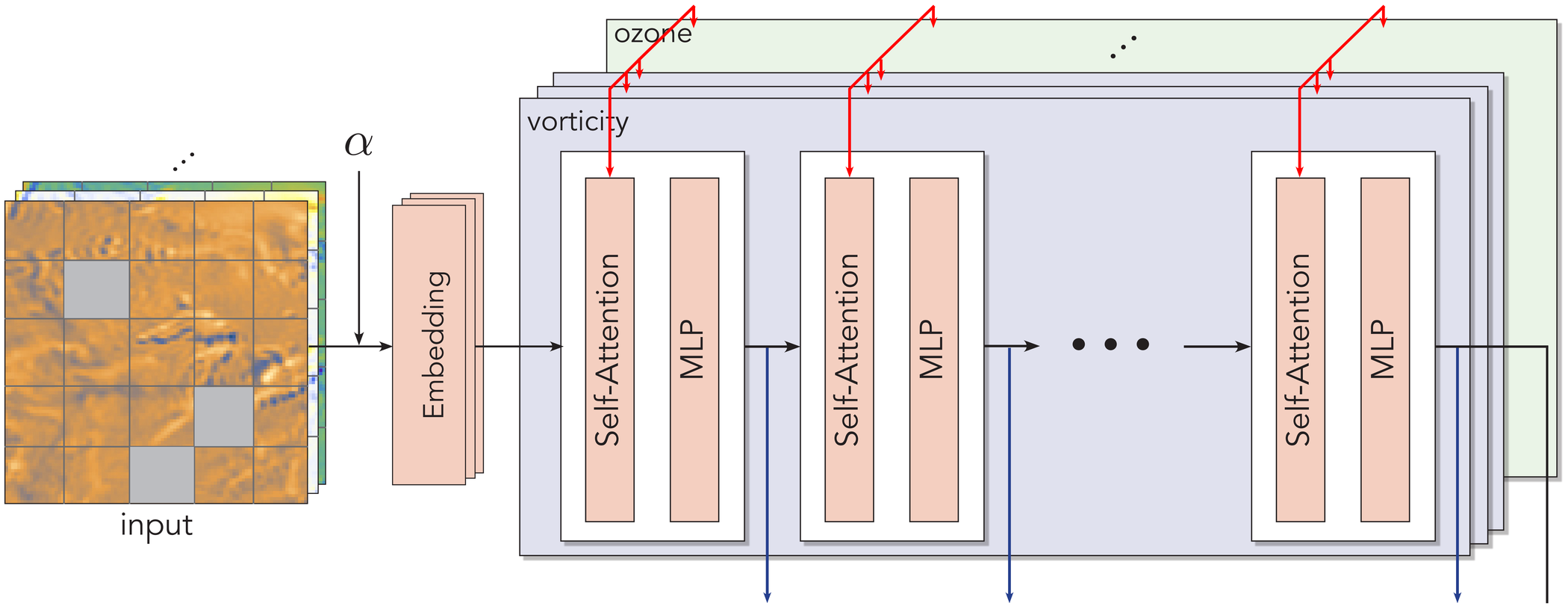
Multiformer



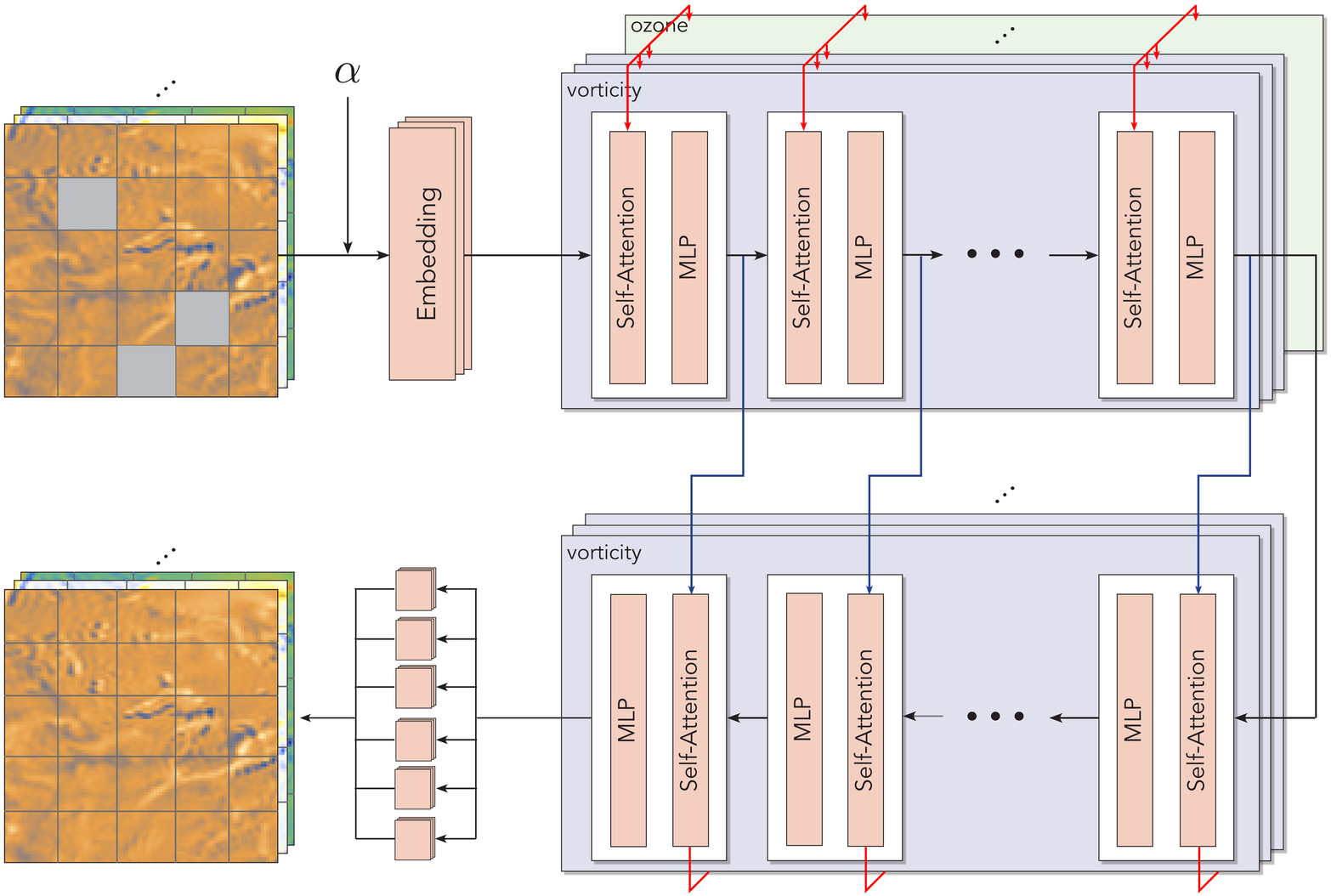
Multiformer



Multiformer



Multiformer



Loss and Pre-training

Training objective

- Numerical statistical atmospheric model

$$p_\theta\left(y|x,\alpha\right)$$

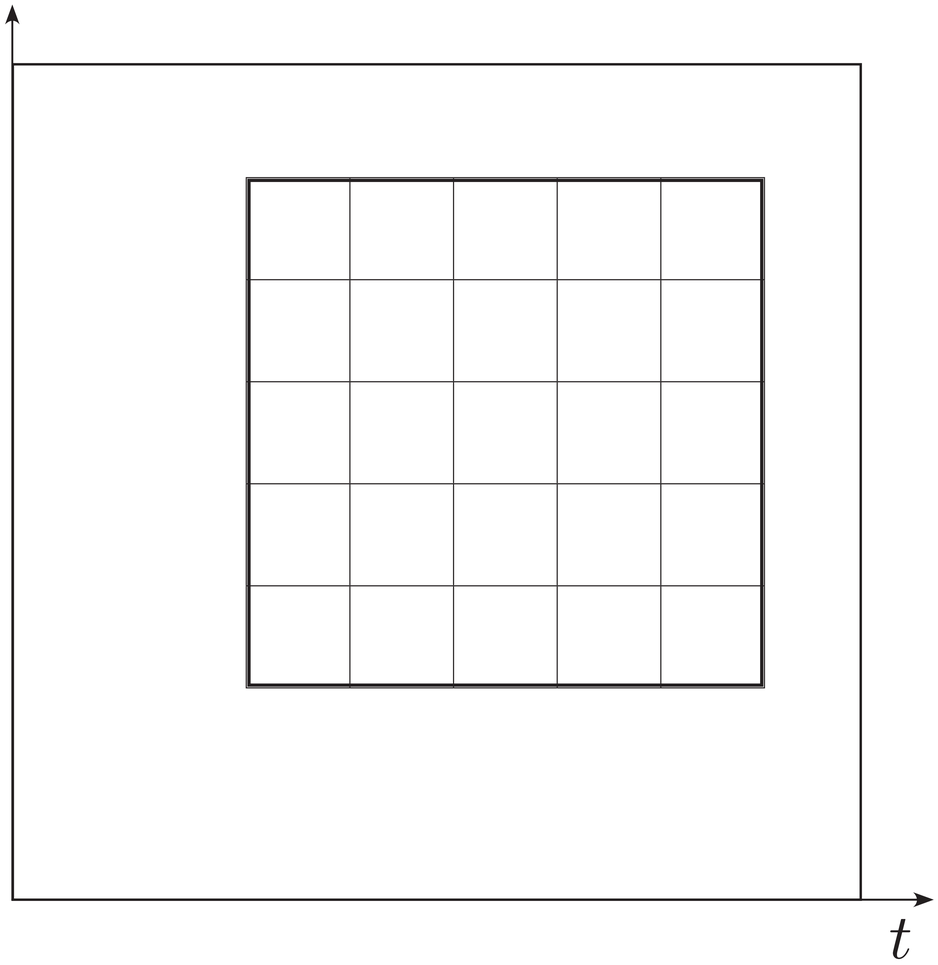
Training objective

- Numerical statistical atmospheric model

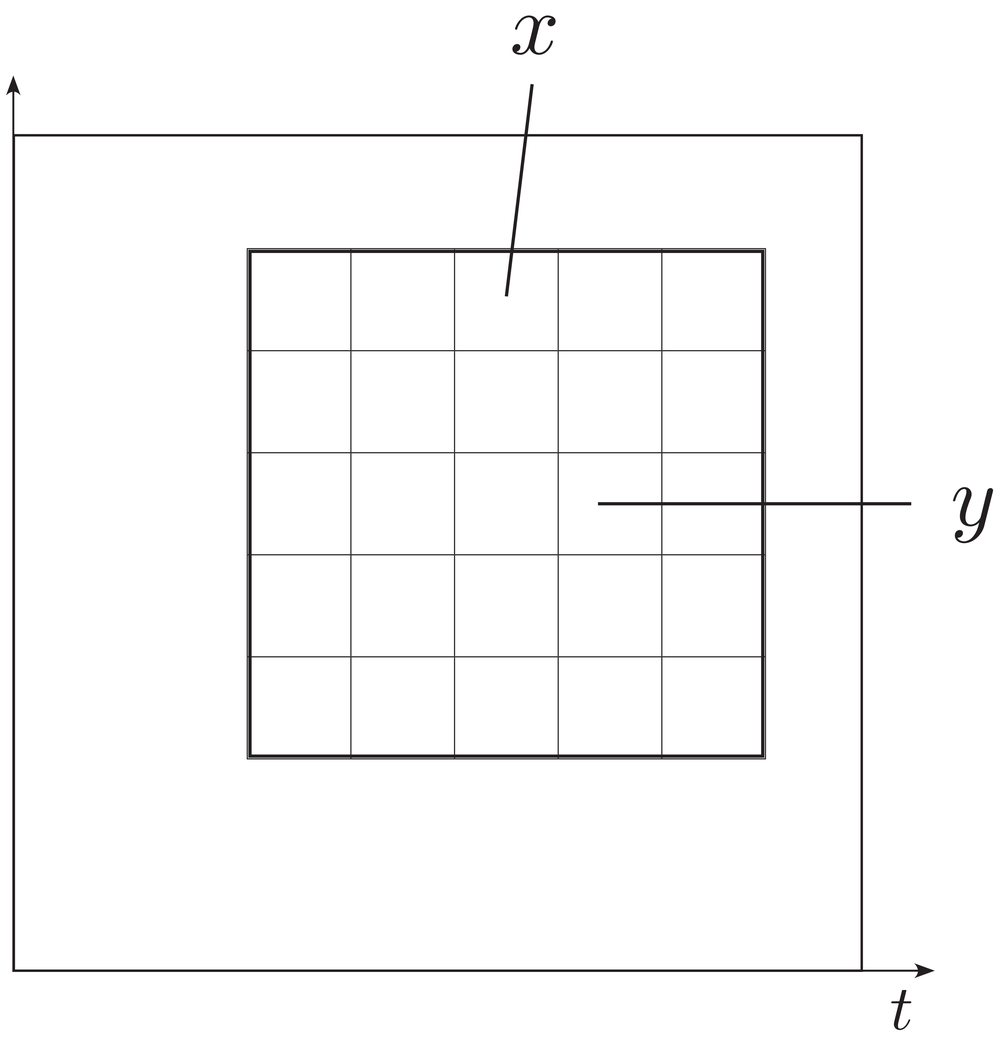
$$p_\theta\left(y|x,\alpha\right)$$

  - Training should model spatio-temporal relationship between arbitrary state x and y

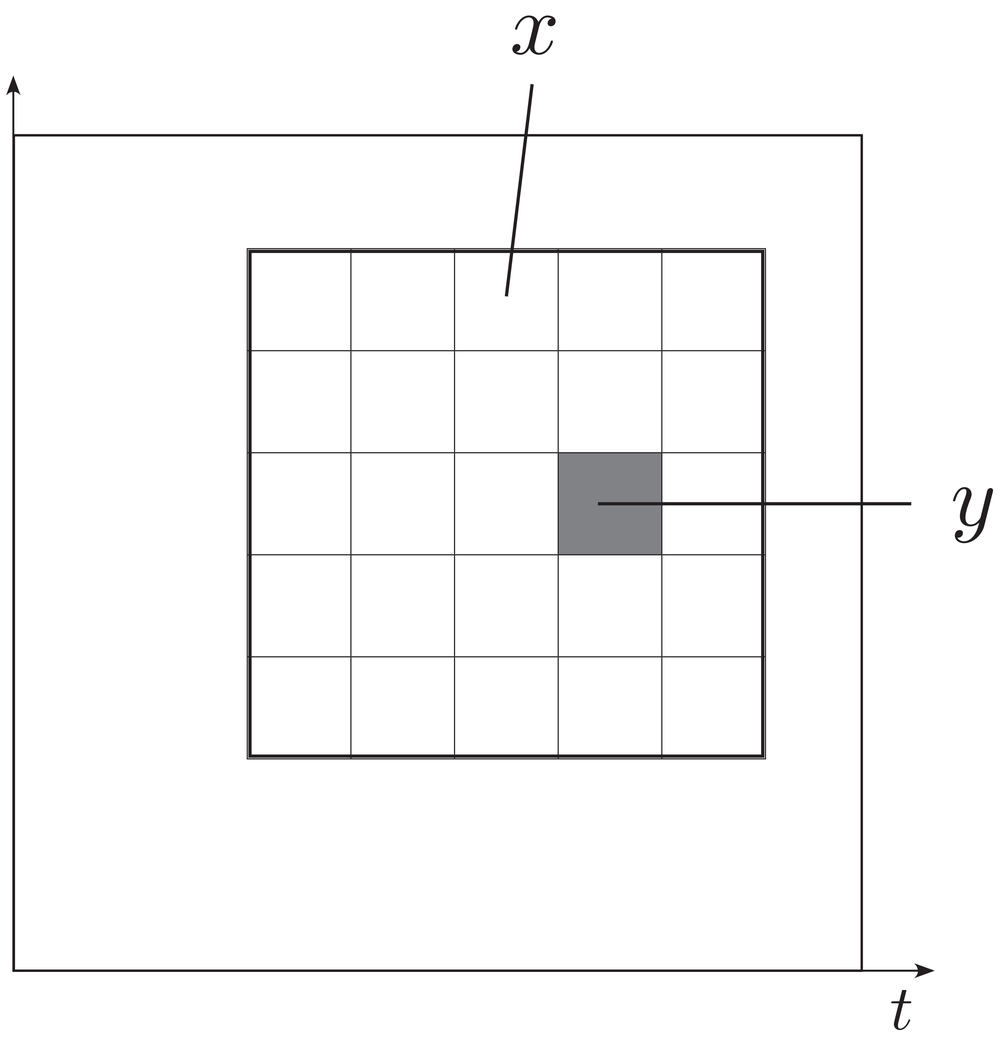
Training objective



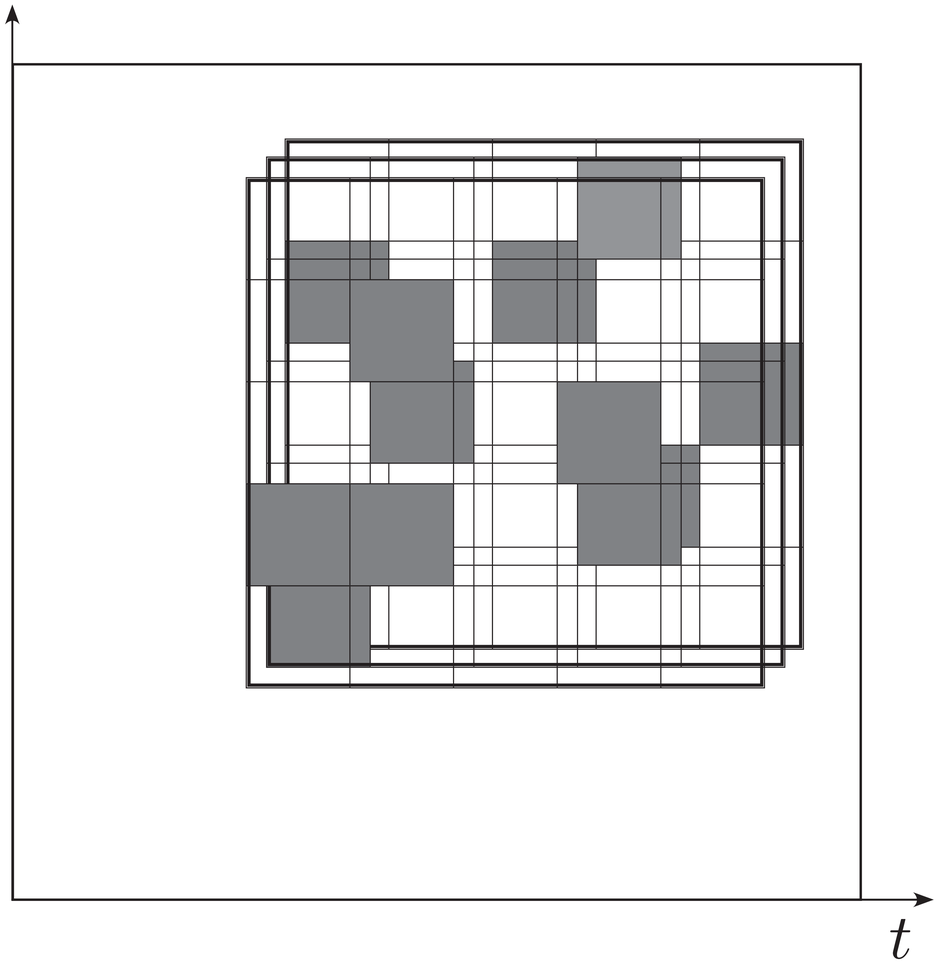
Training objective



Training objective



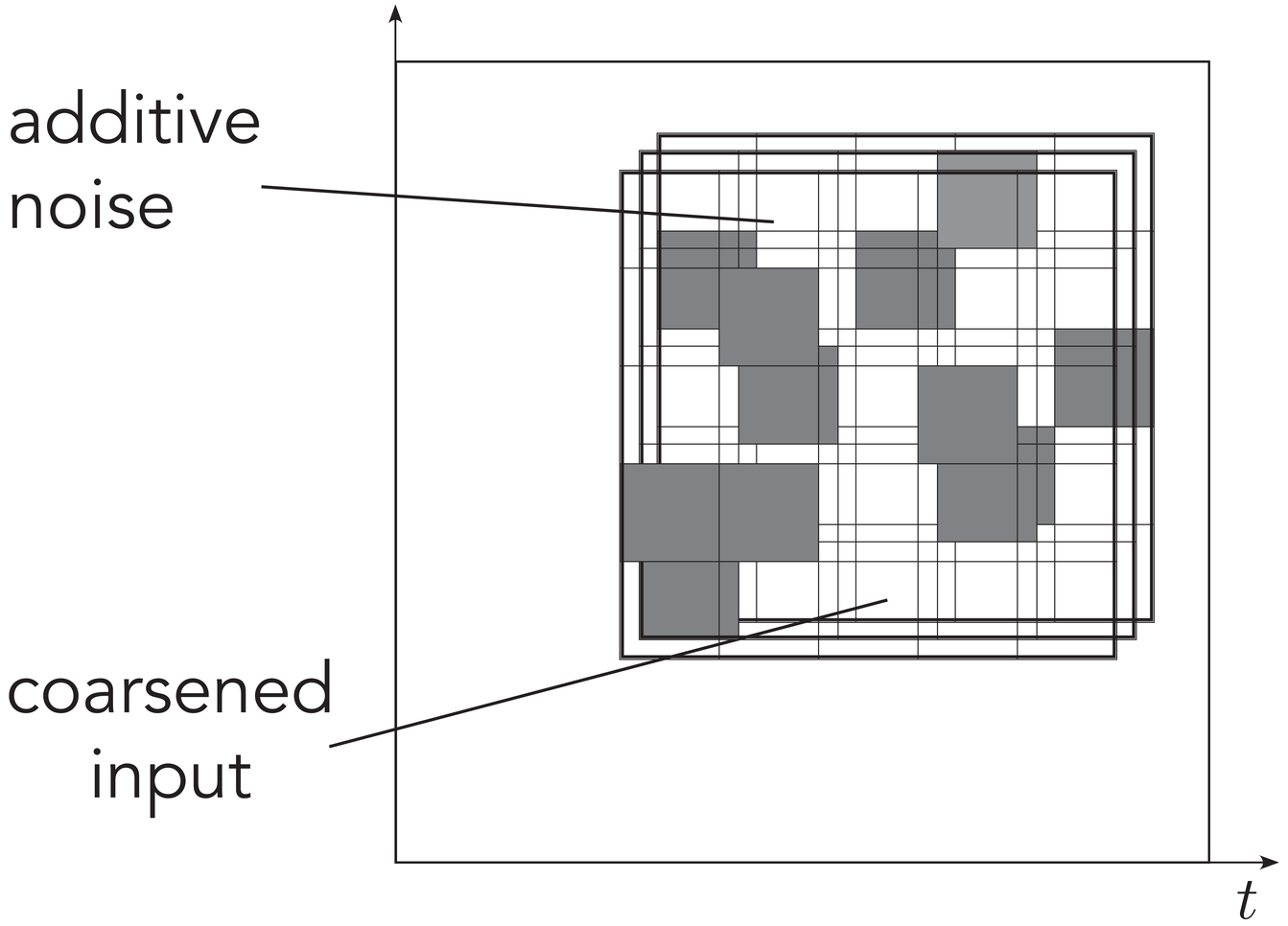
Training objective



masked
token model:

training to predict
randomly masked
information

Training objective

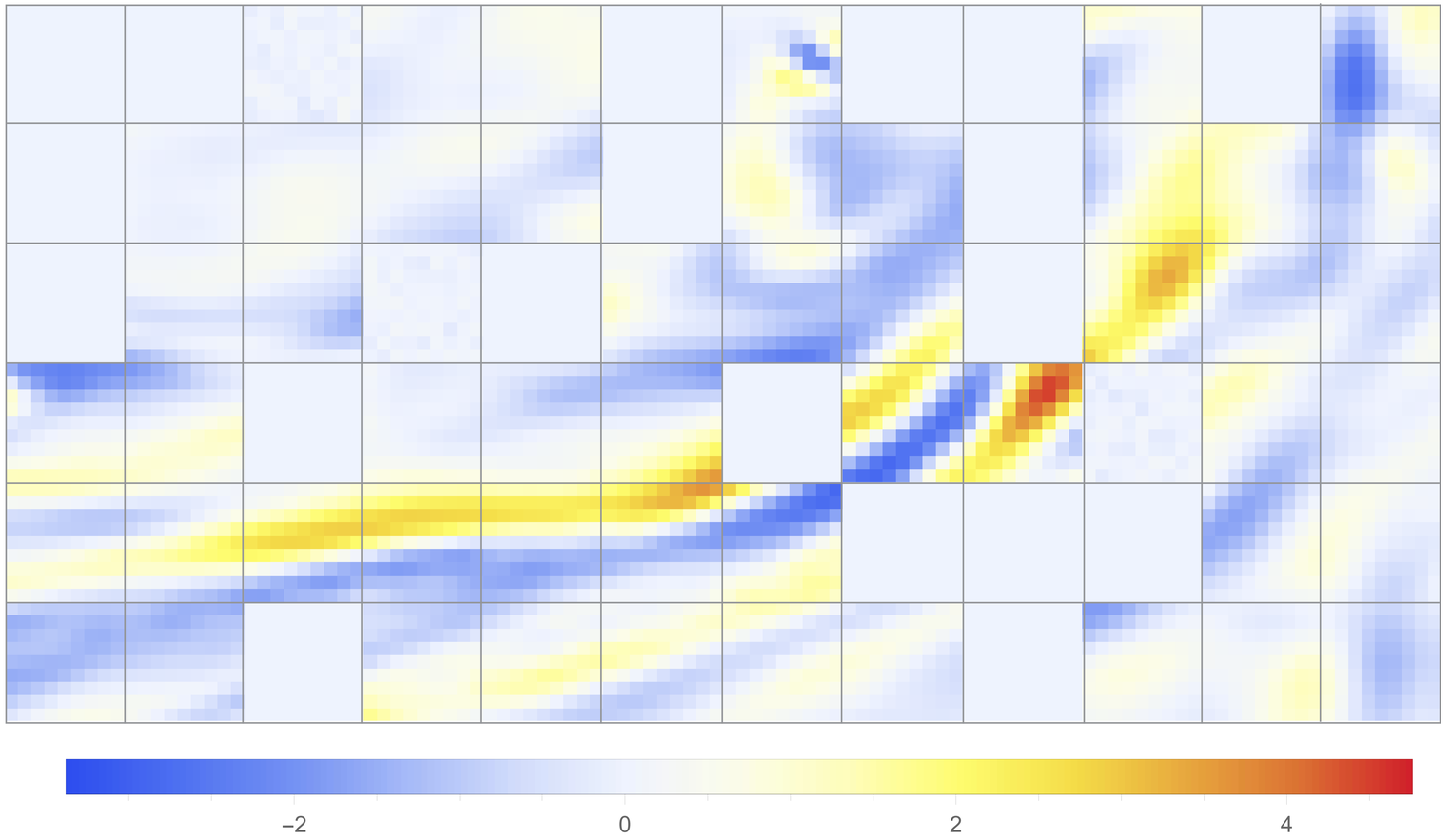


masked
token model:

training to predict
randomly masked
information

Training objective

divergence, ml=96



Loss

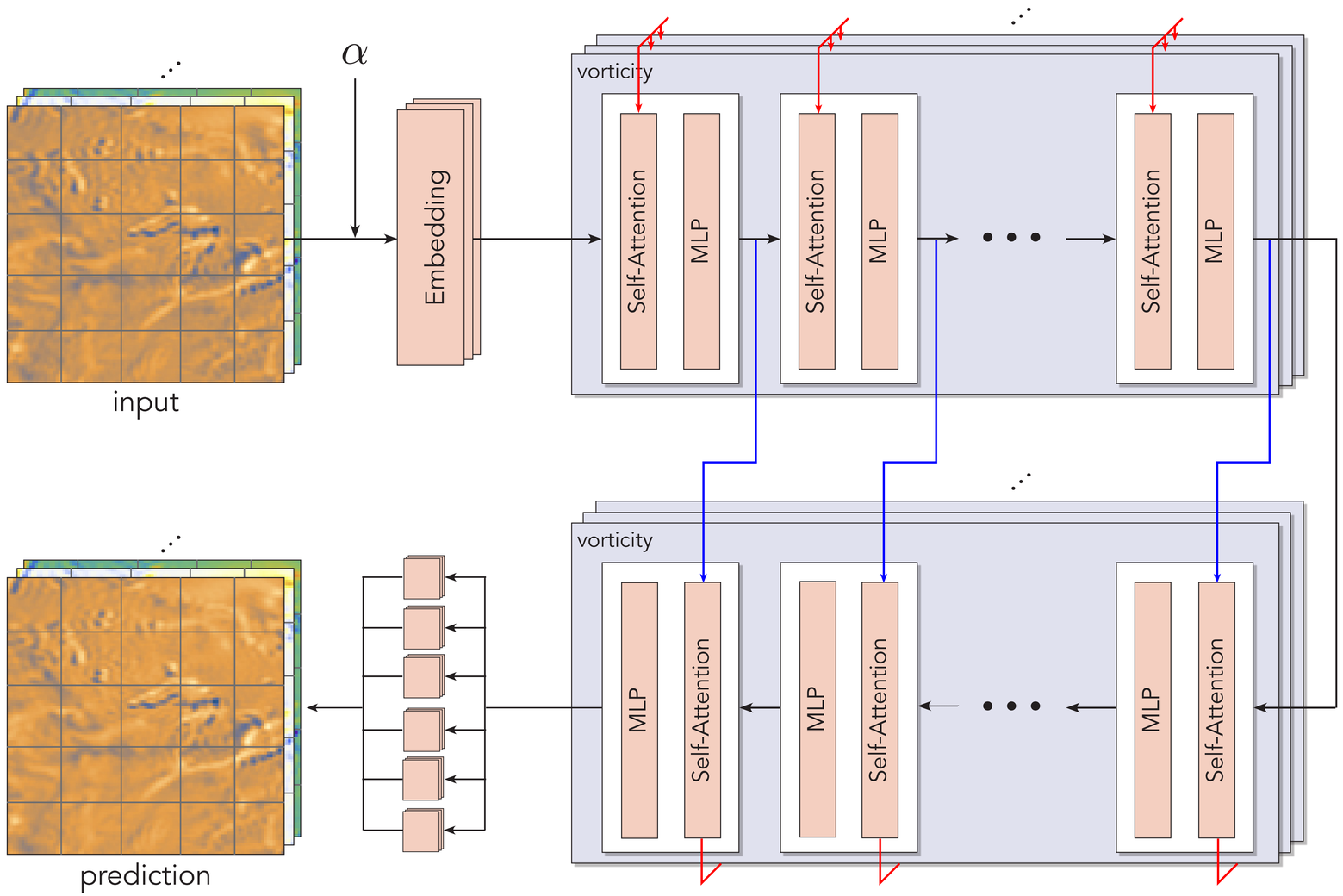
- MSE measures only very approximately what is of interest in applications
- To train a statistical model, deterministic training and loss is insufficient

Loss

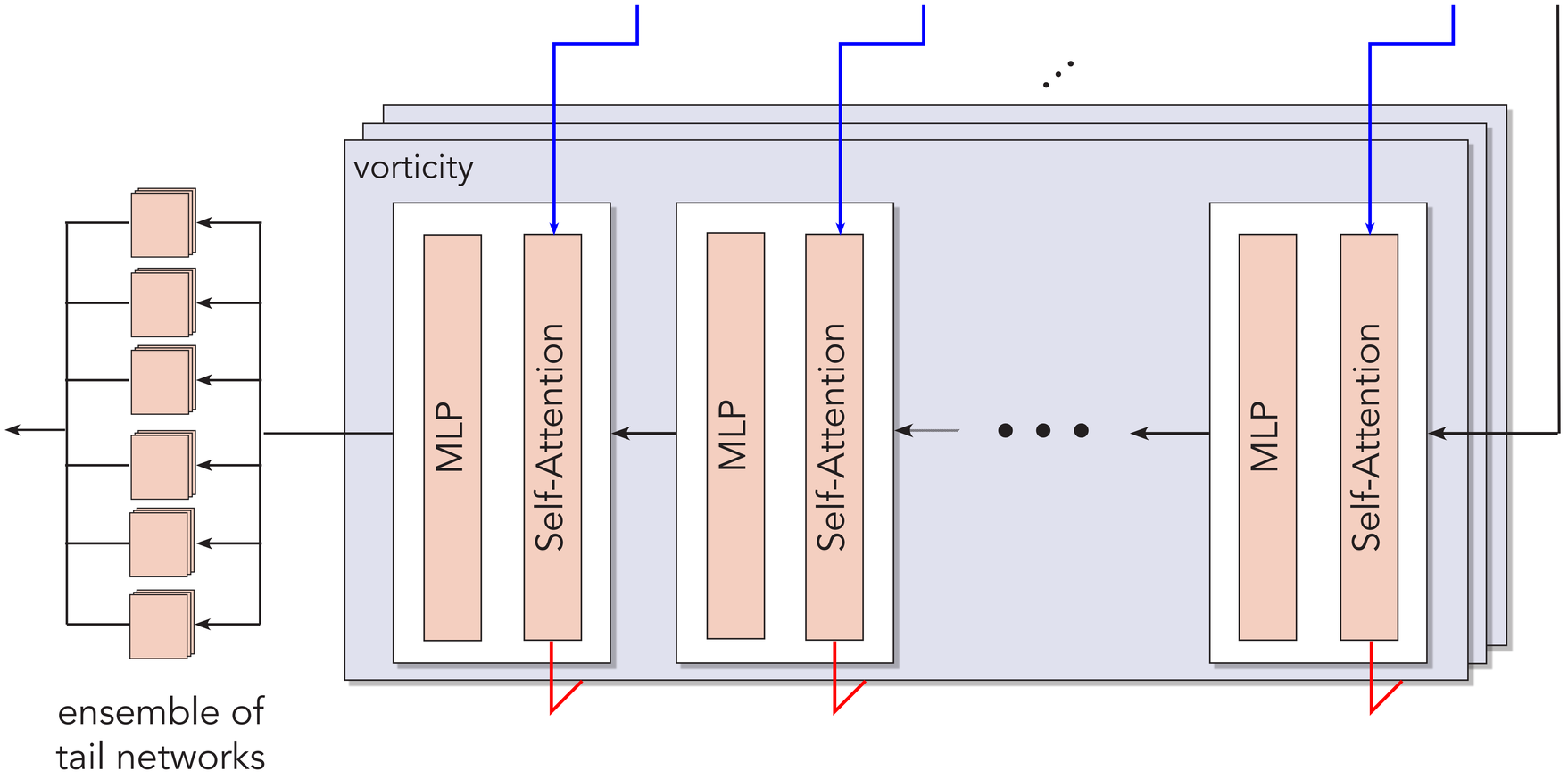
- MSE measures only very approximately what is of interest in applications
- To train a statistical model, deterministic training and loss is insufficient

=> Analog of cross-entropy loss for regression problems?

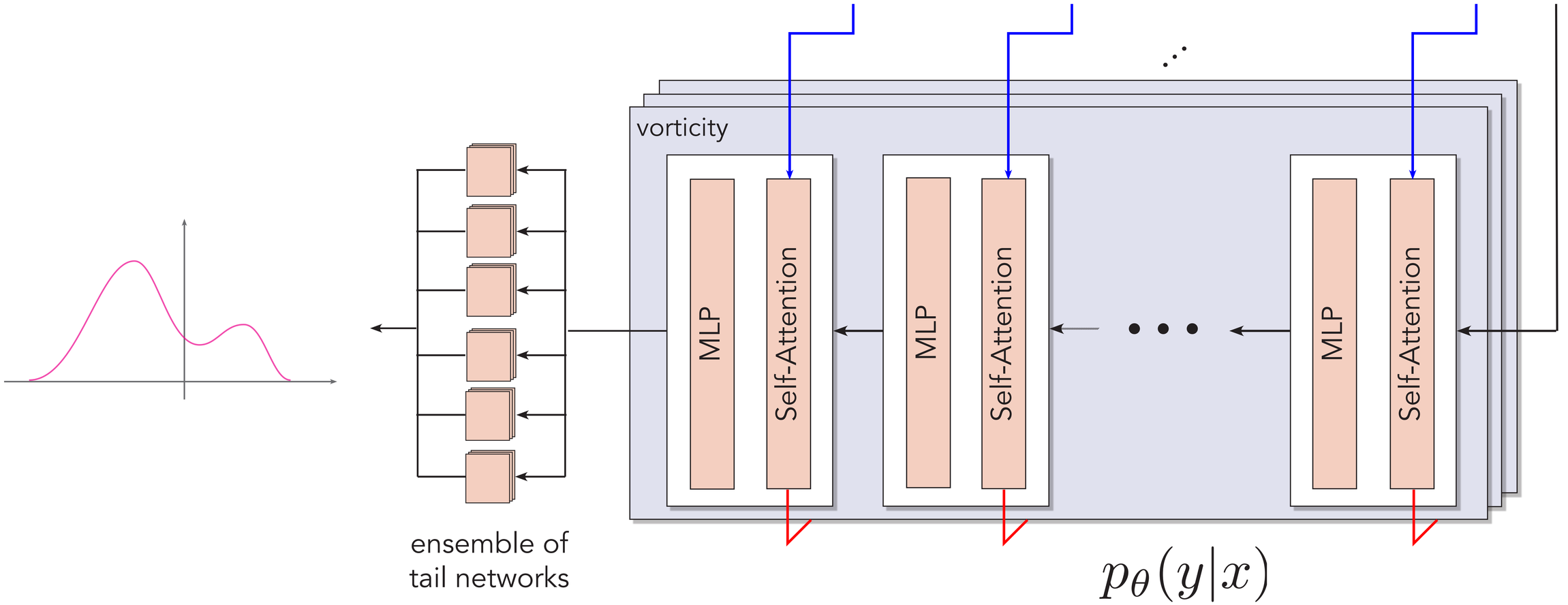
Statistical loss



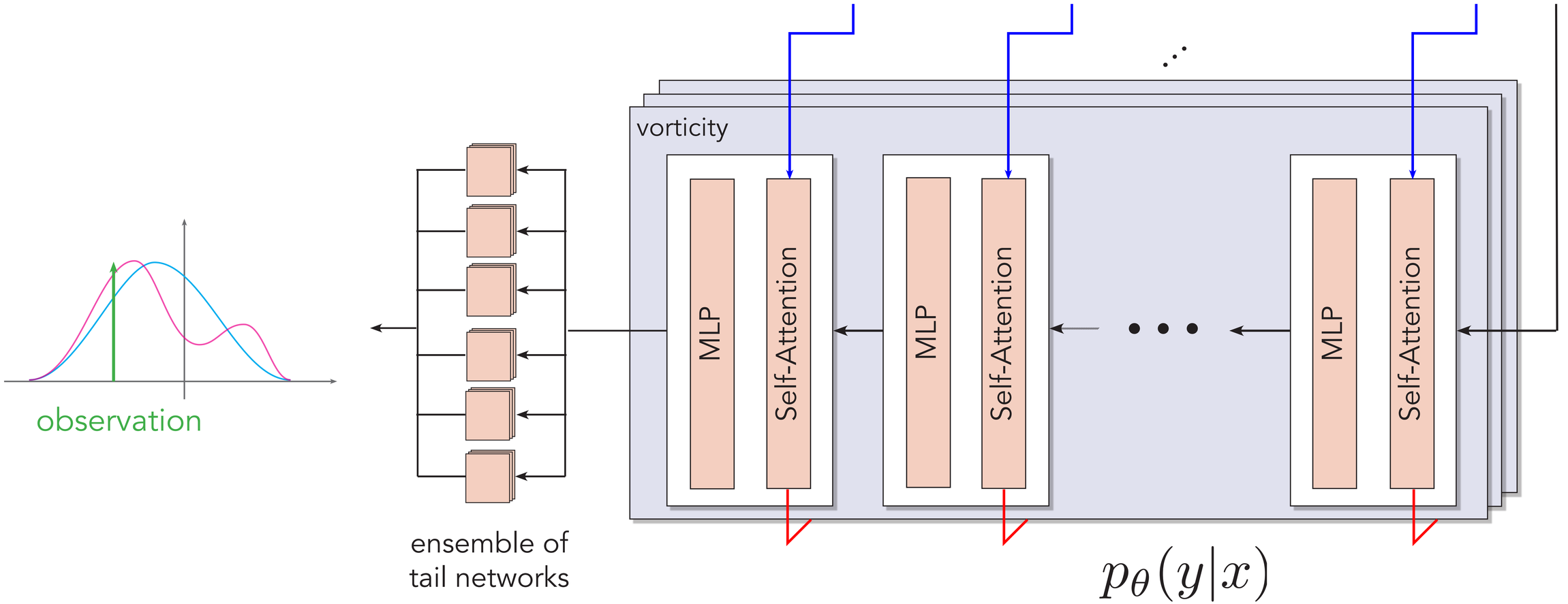
Statistical loss



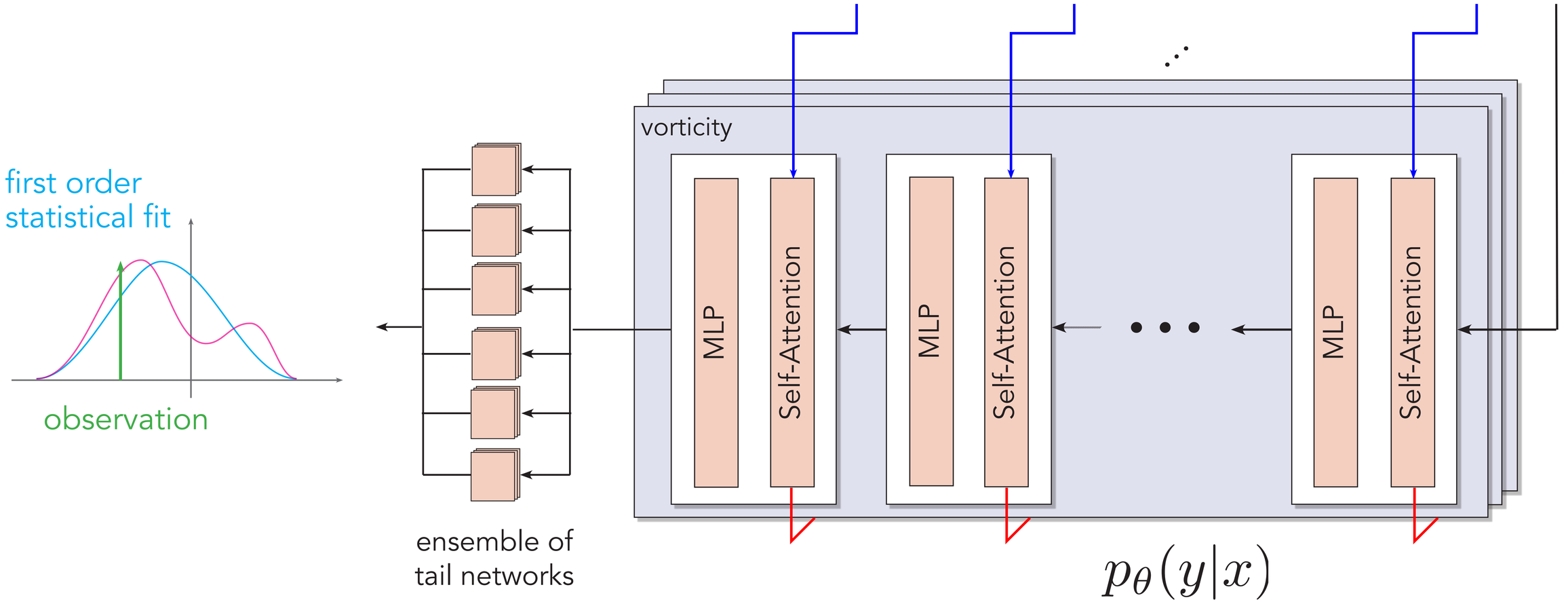
Statistical loss



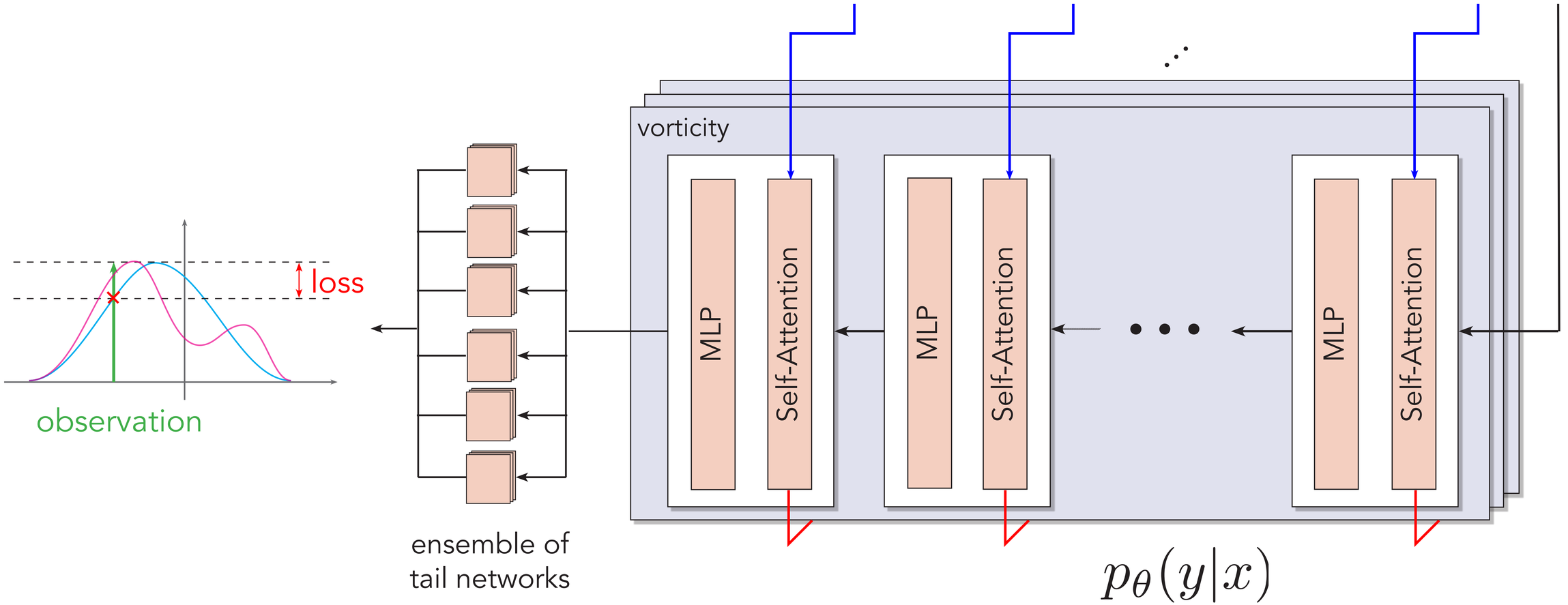
Statistical loss



Statistical loss

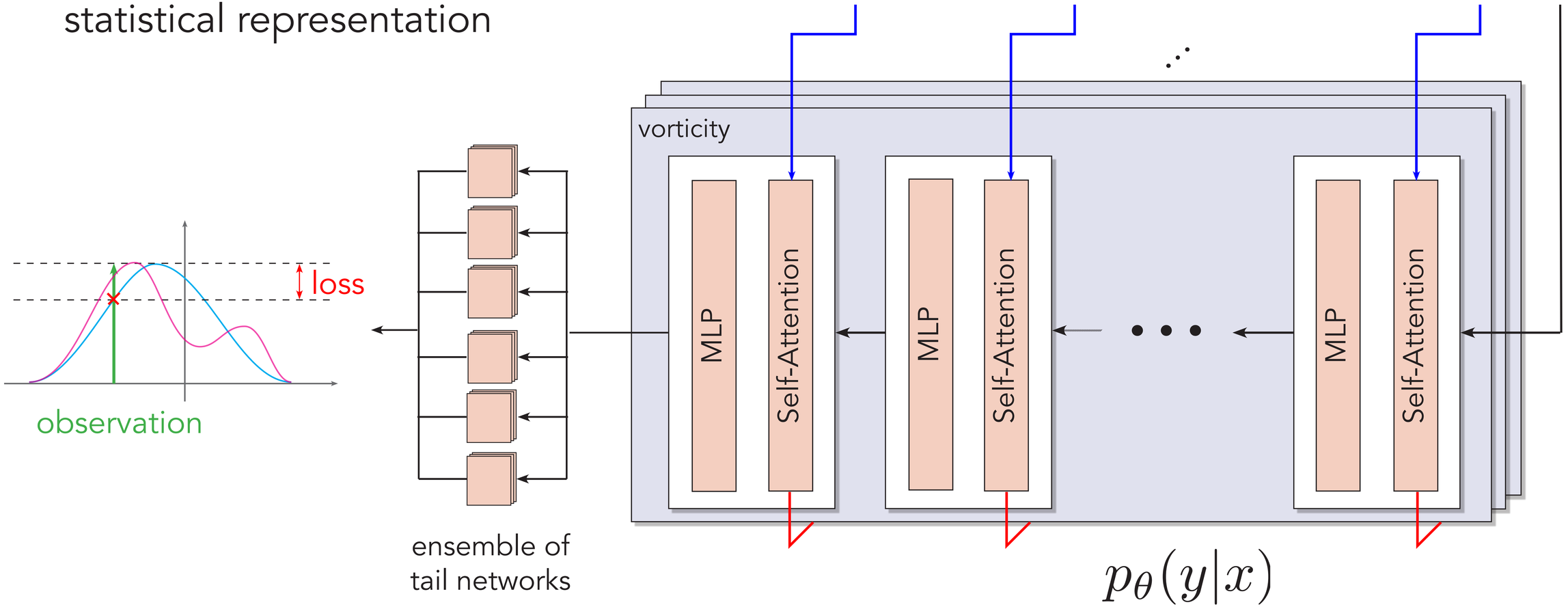


Statistical loss

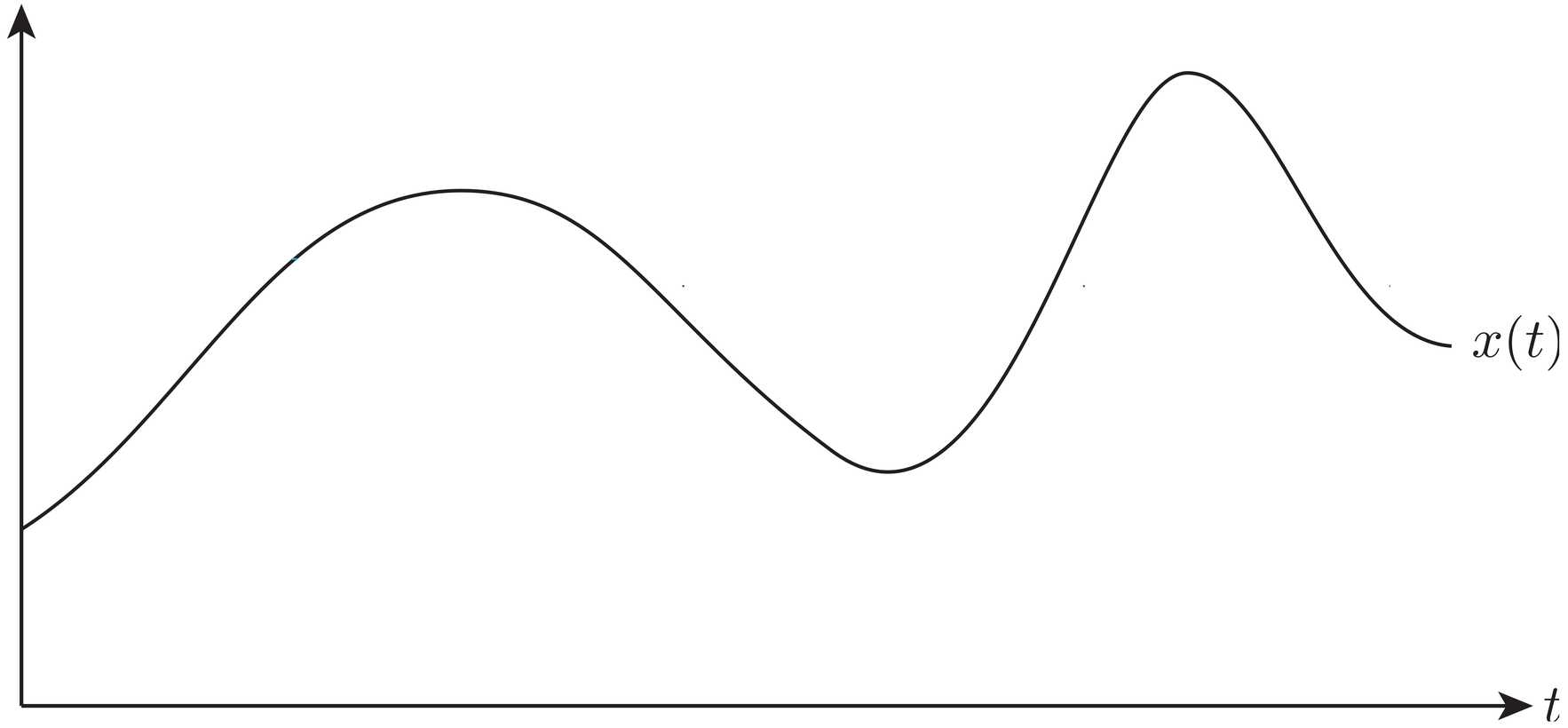


Statistical loss

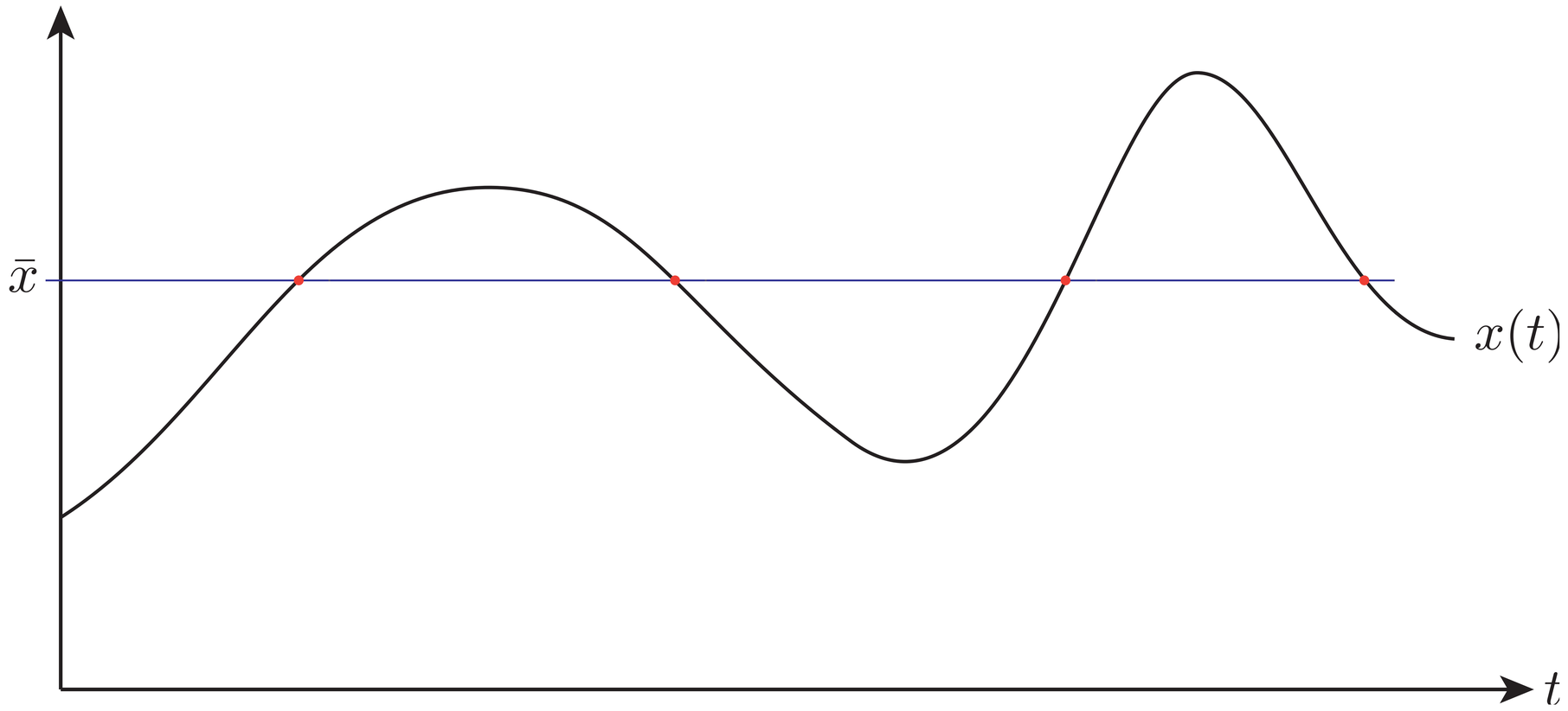
end-to-end training encourages
statistical representation



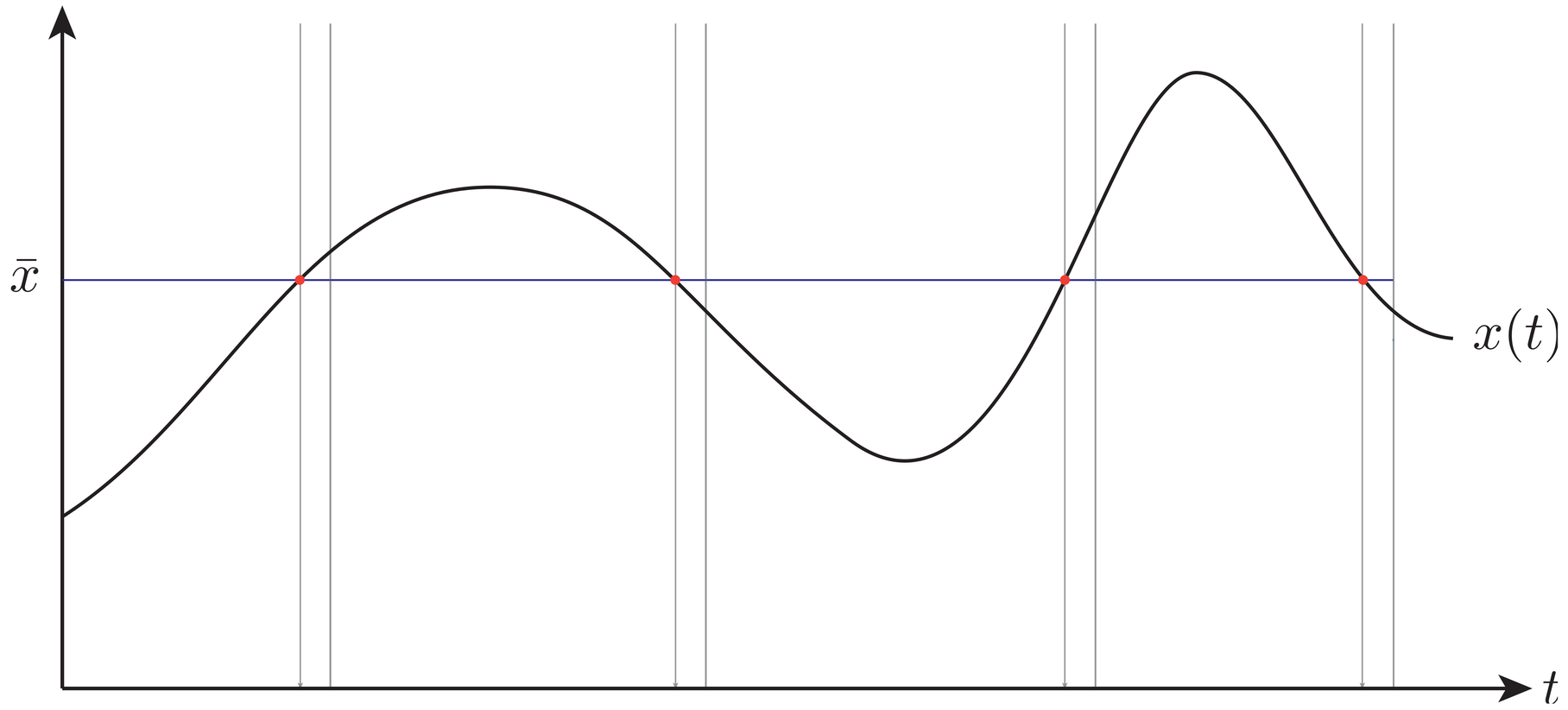
Statistical loss



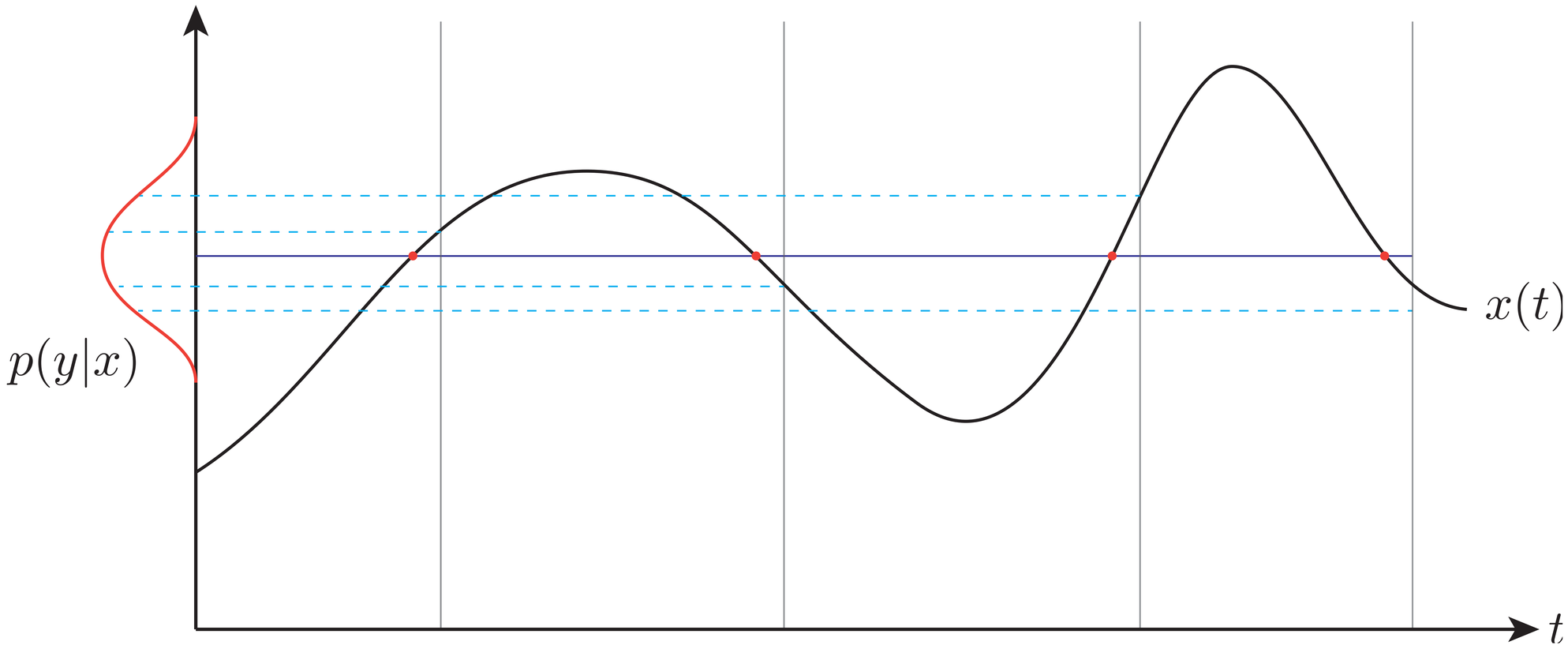
Statistical loss



Statistical loss

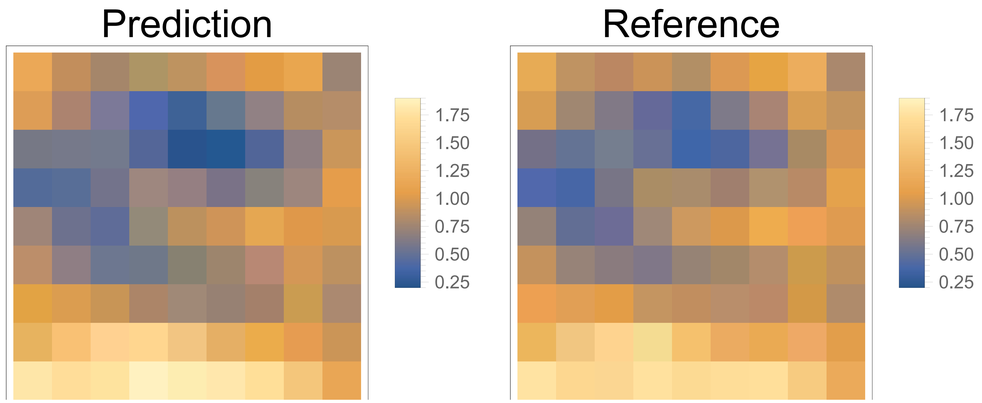


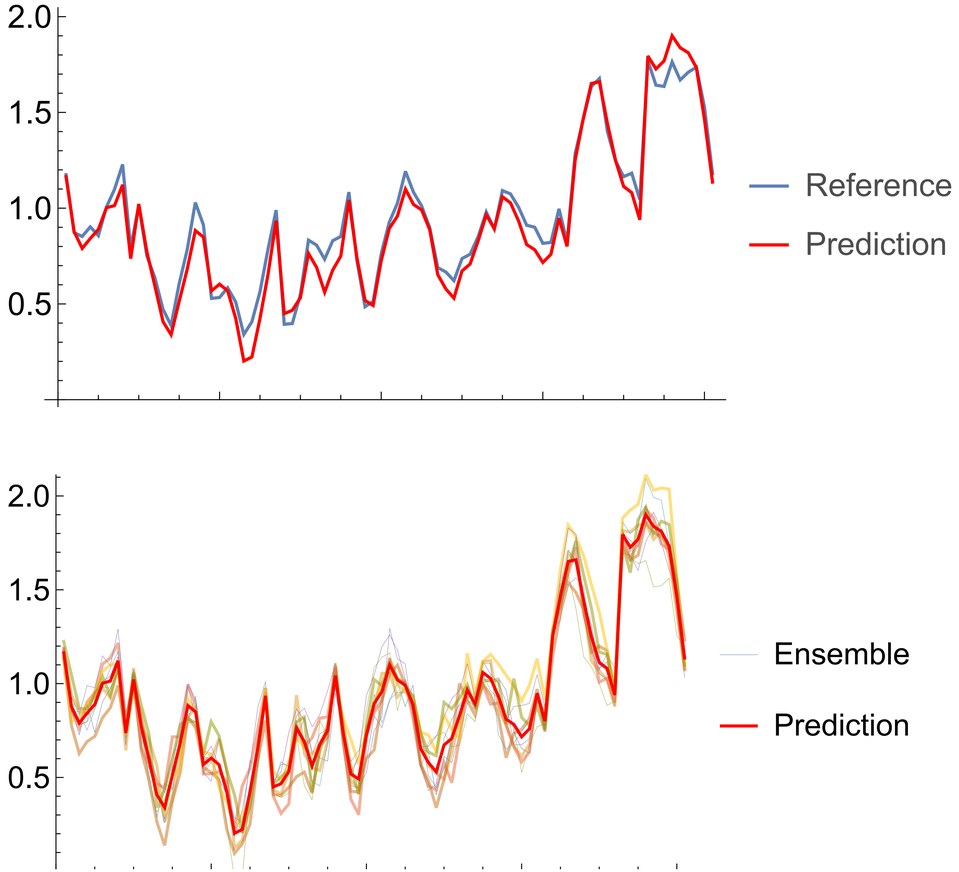
Statistical loss



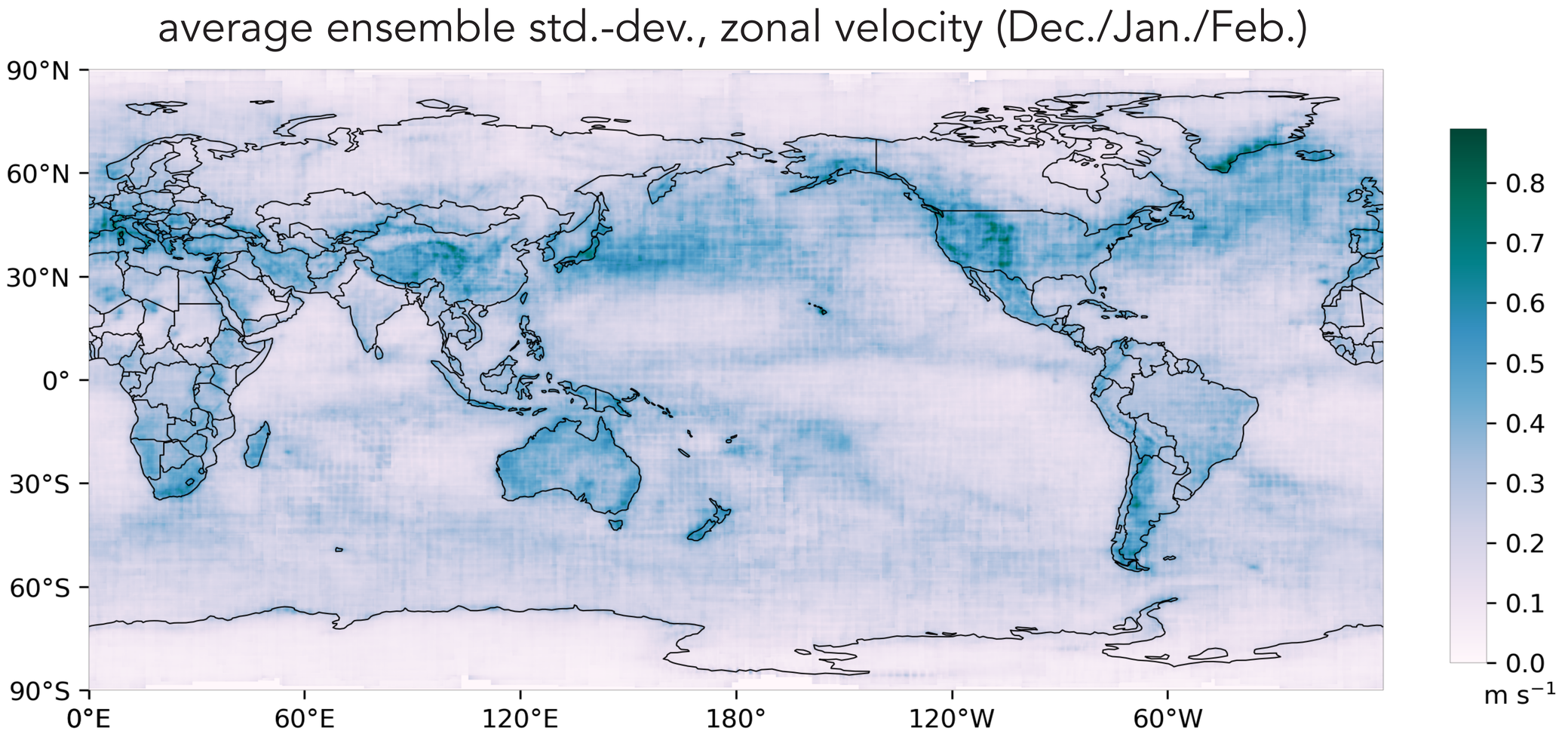
Statistical loss

- Predictions:

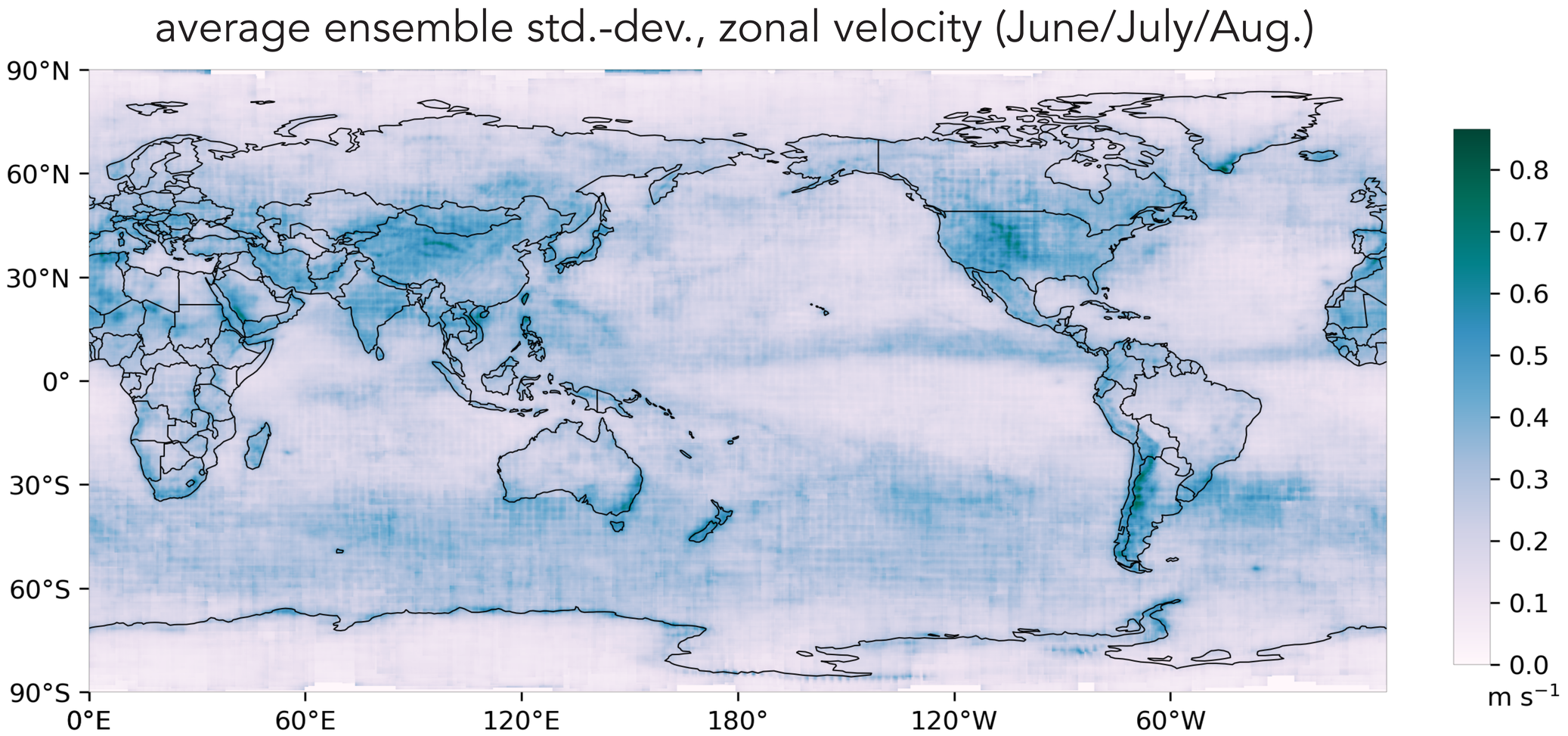




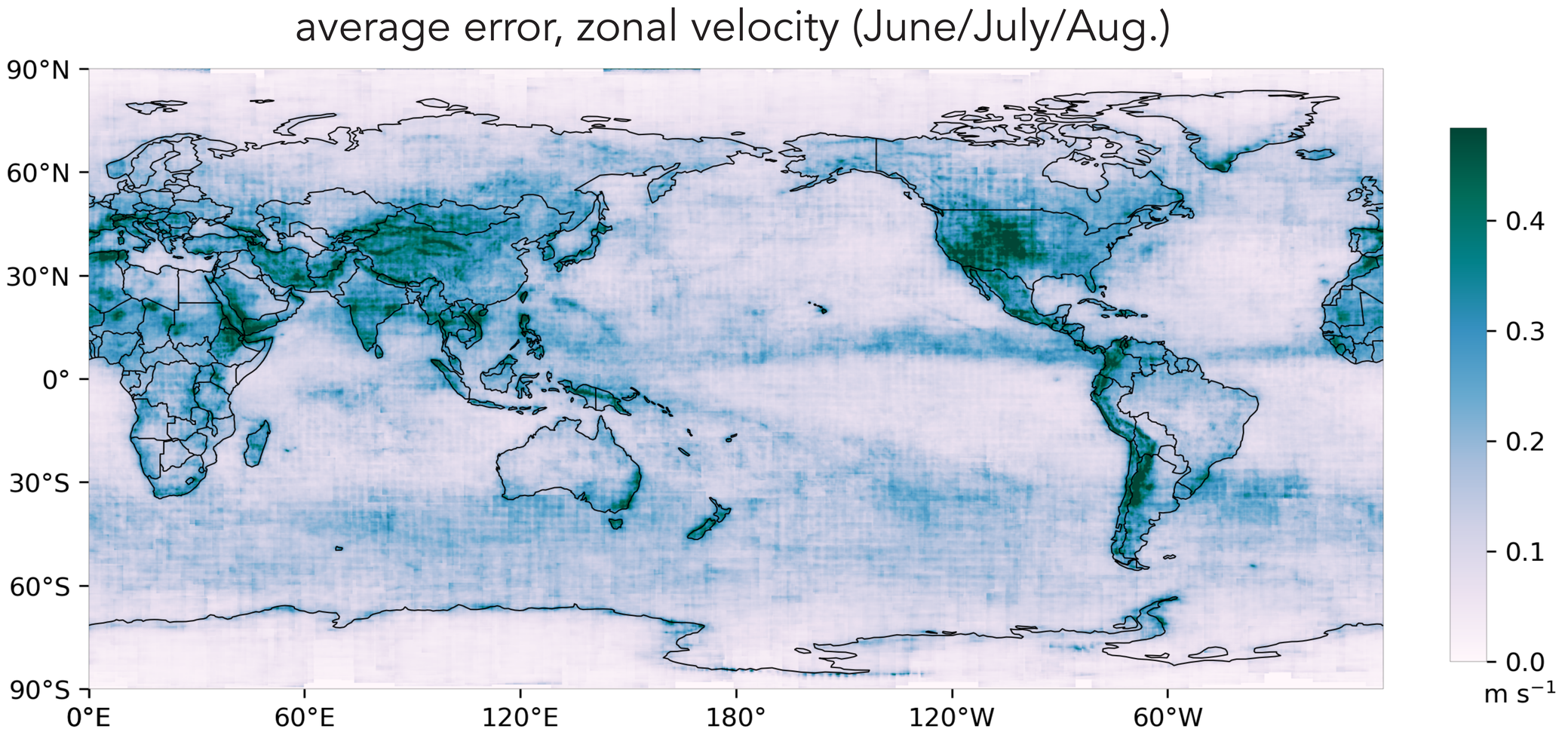
Statistical loss



Statistical loss



Statistical loss



Intrinsic Capabilities

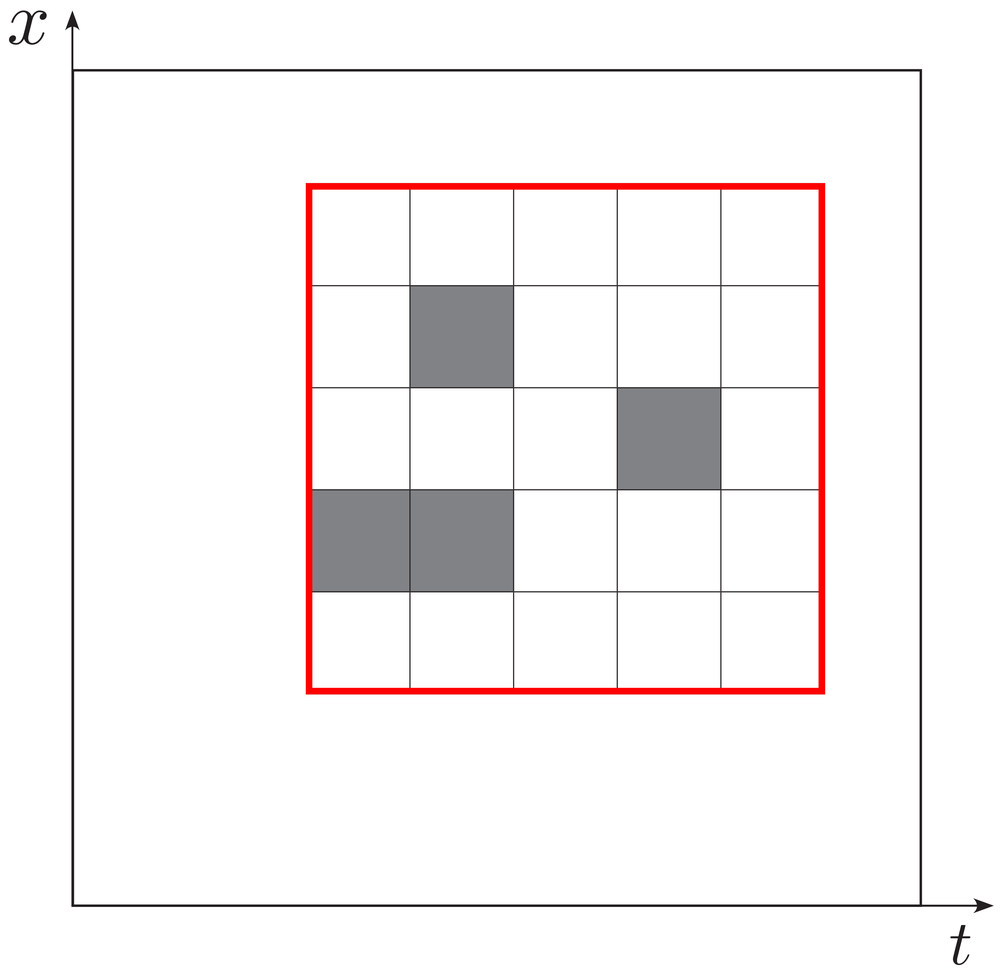
Zero-shot capabilities

- Numerical statistical atmospheric model:

$$p_\theta\left(y|x,\alpha\right)$$

- Model directly includes important applications: forecasting, downscaling, temporal interpolation, ...

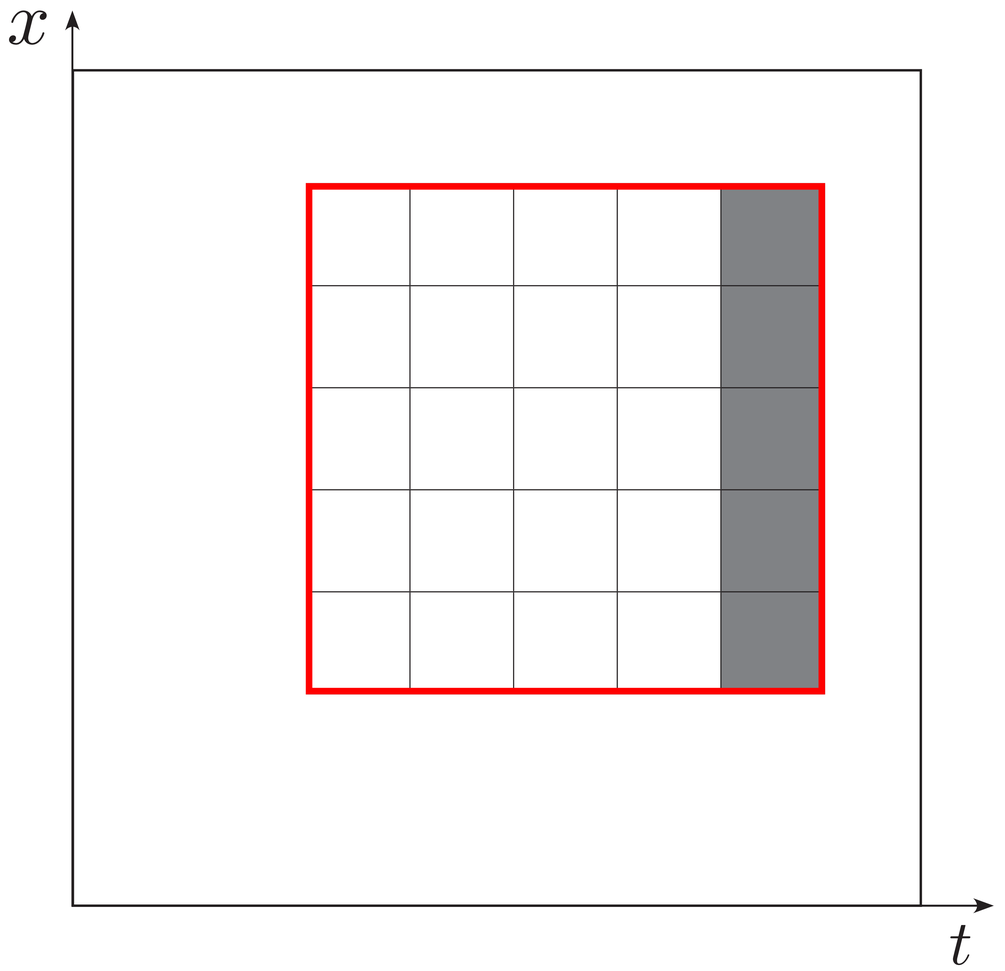
Zero-shot capabilities



Training task:

predict randomly masked neighbor-hoods in space-time

Zero-shot capabilities

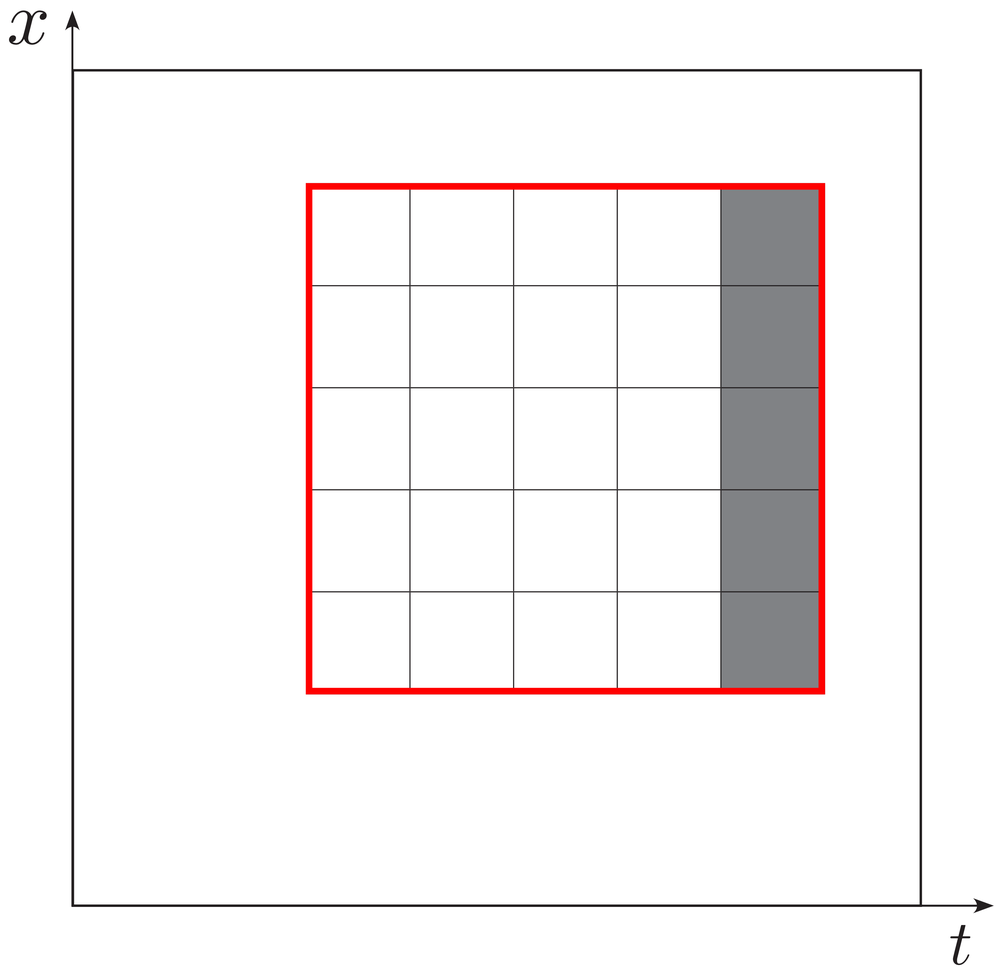


Training task:

predict randomly masked neighbor-hoods in space-time

Zero-shot capabilities

Forecasting

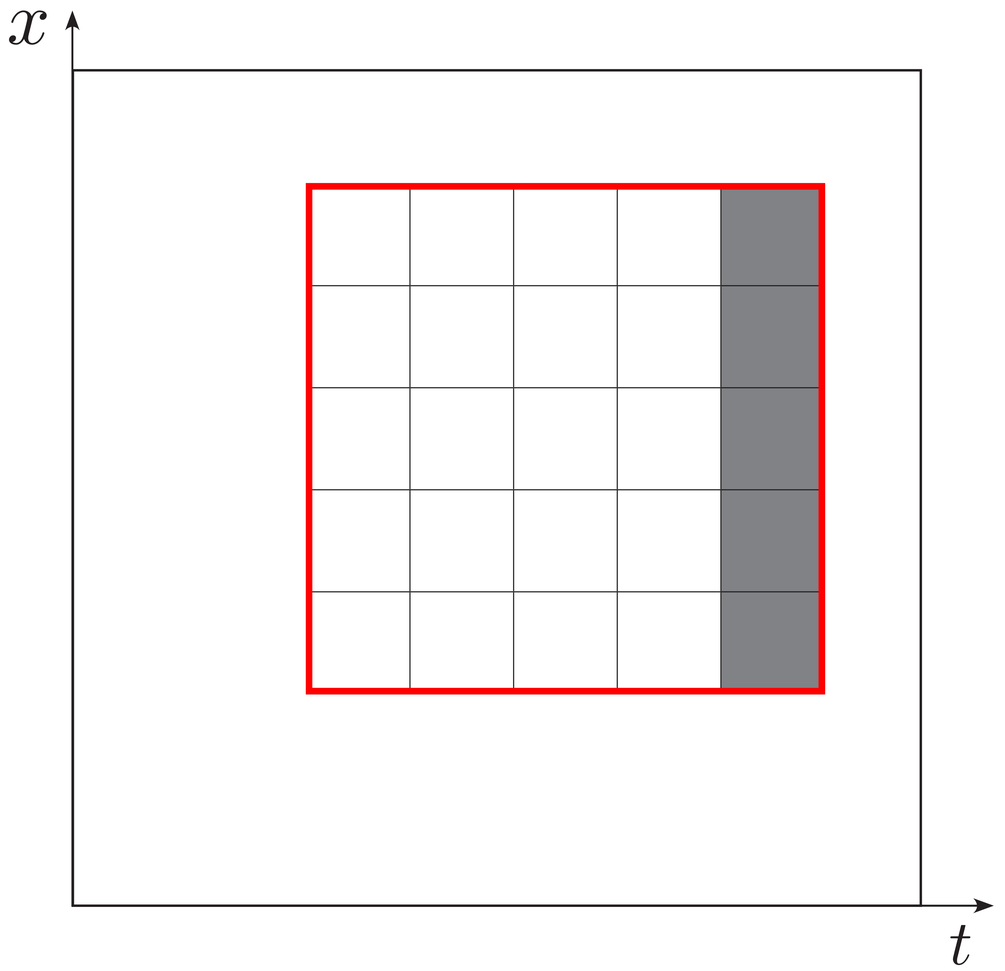


Training task:

predict randomly masked neighbor-hoods in space-time

Zero-shot capabilities

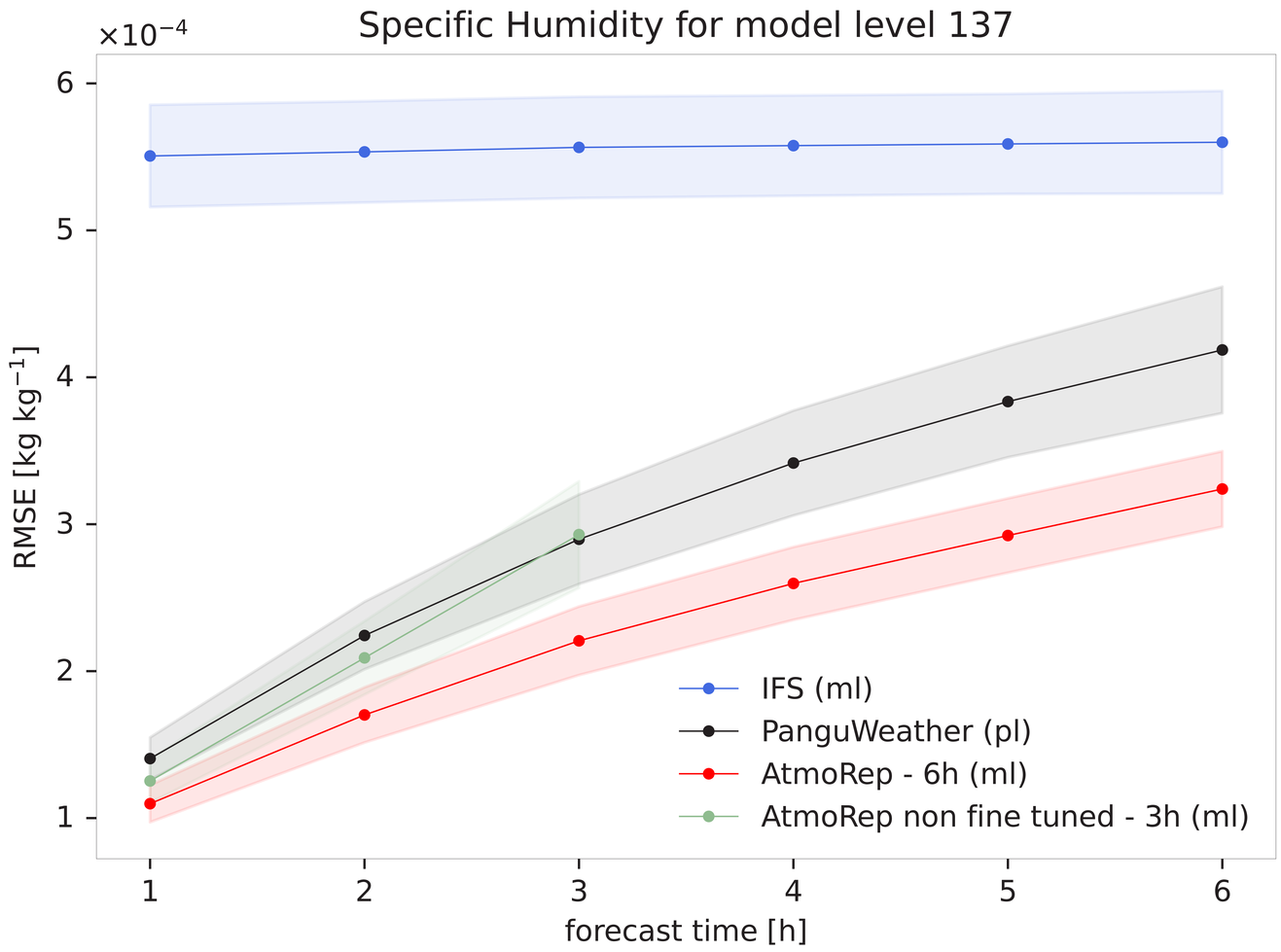
Forecasting



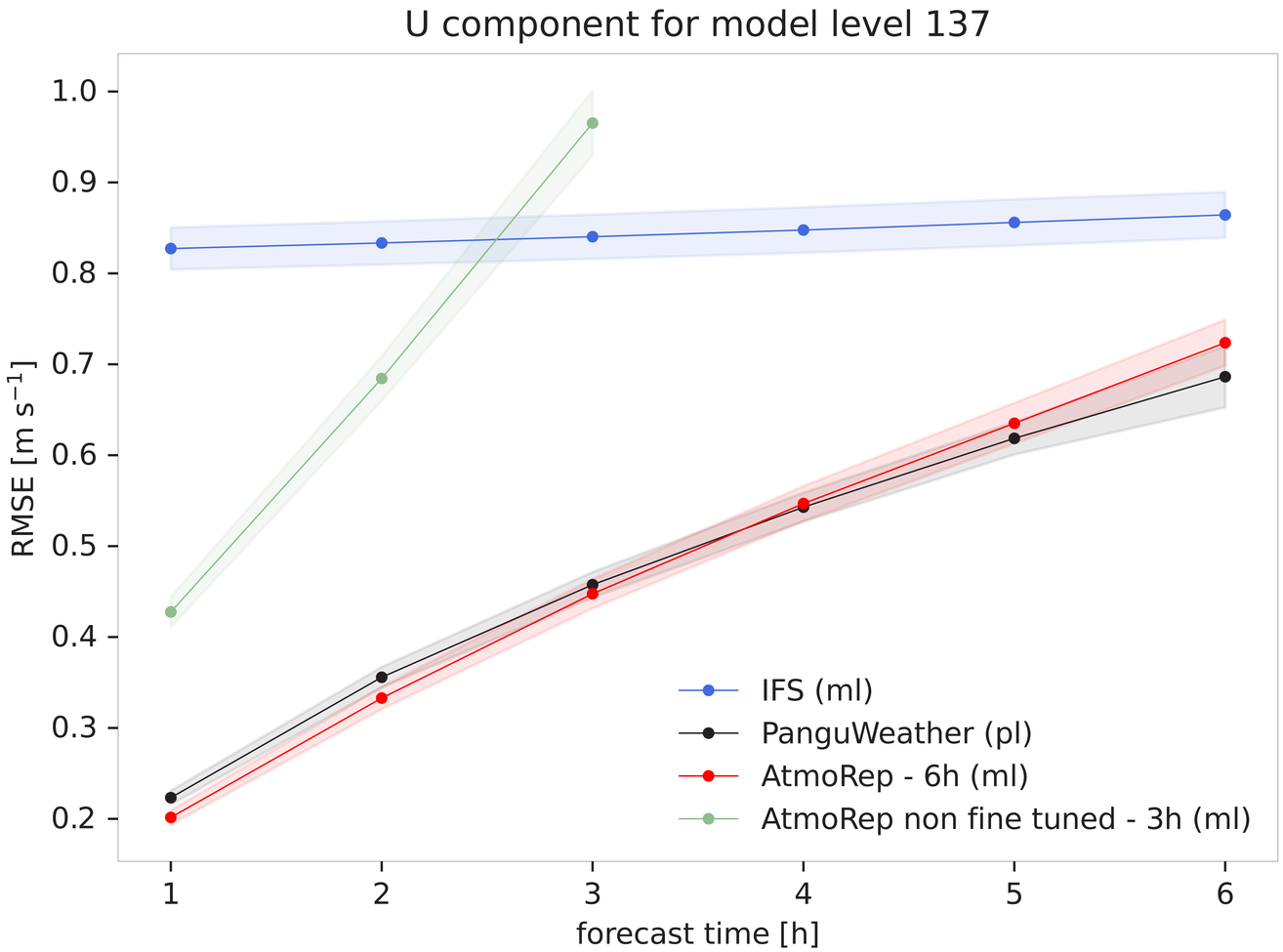
Training task:

predict randomly masked neighbor-hoods in space-time

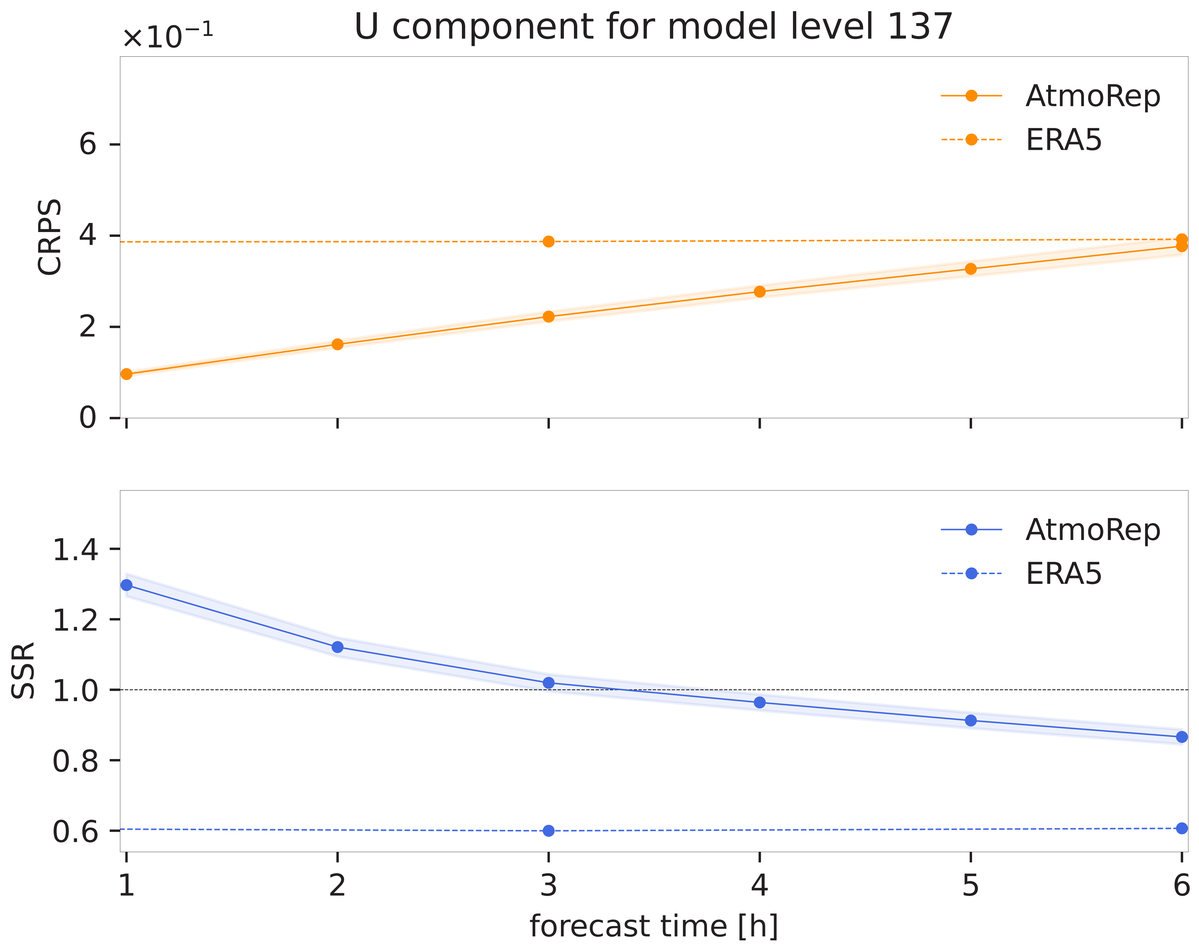
Zero-shot capabilities



Zero-shot capabilities

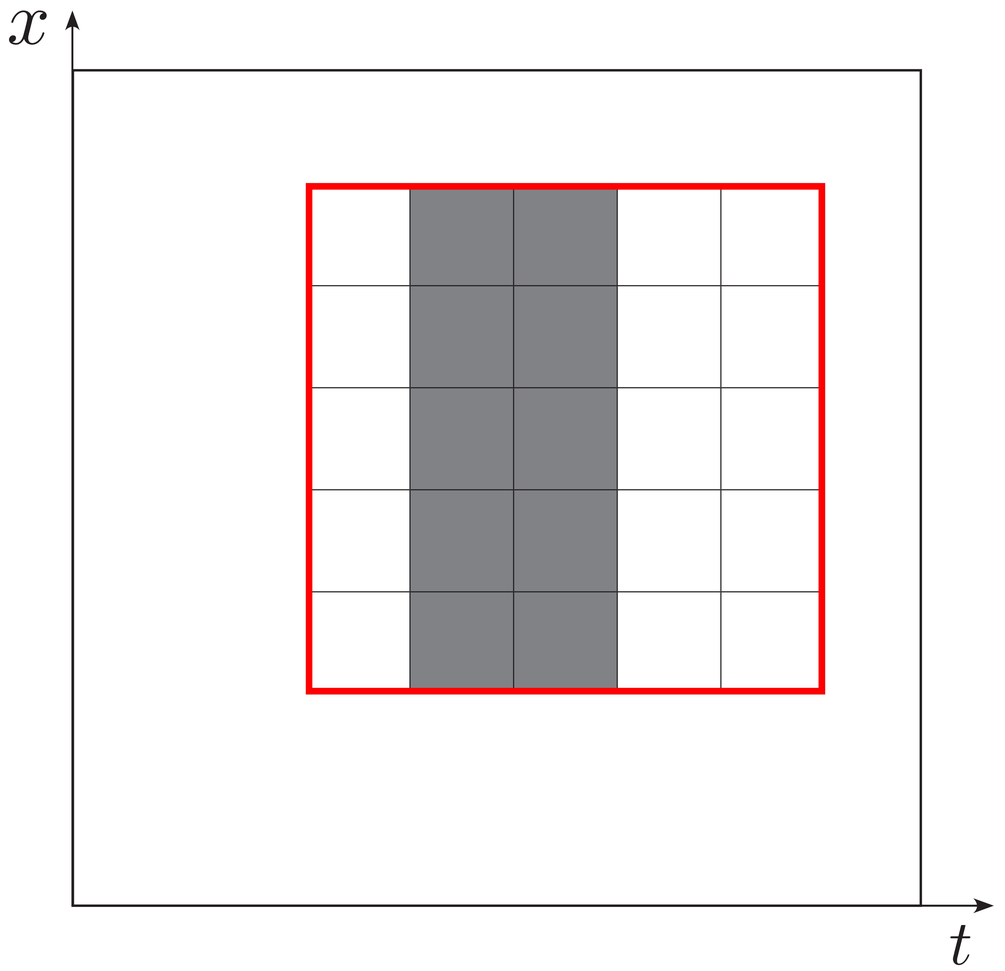


Zero-shot capabilities



Zero-shot capabilities

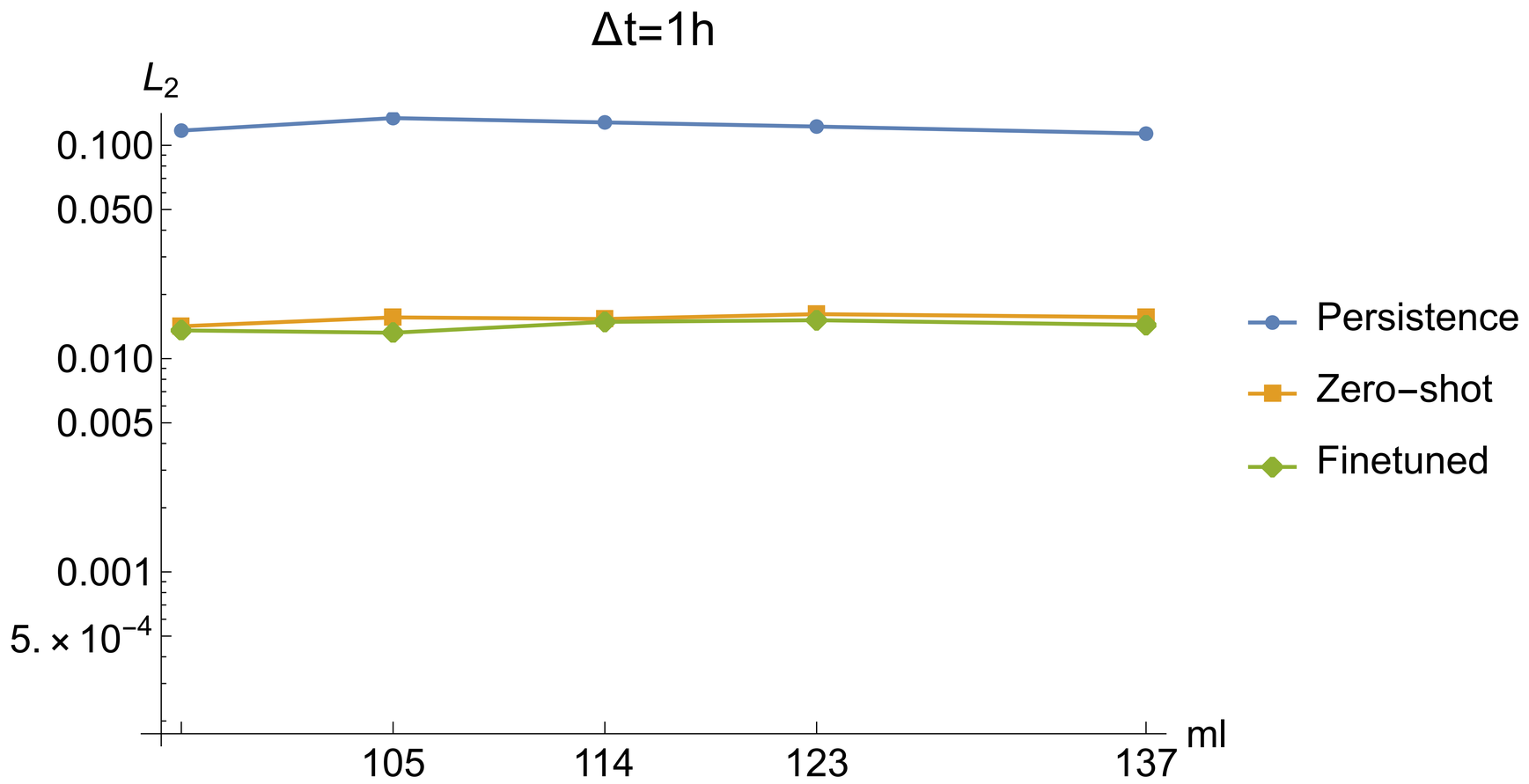
Temporal interpolation



Training task:

predict randomly masked neighbor-hoods in space-time

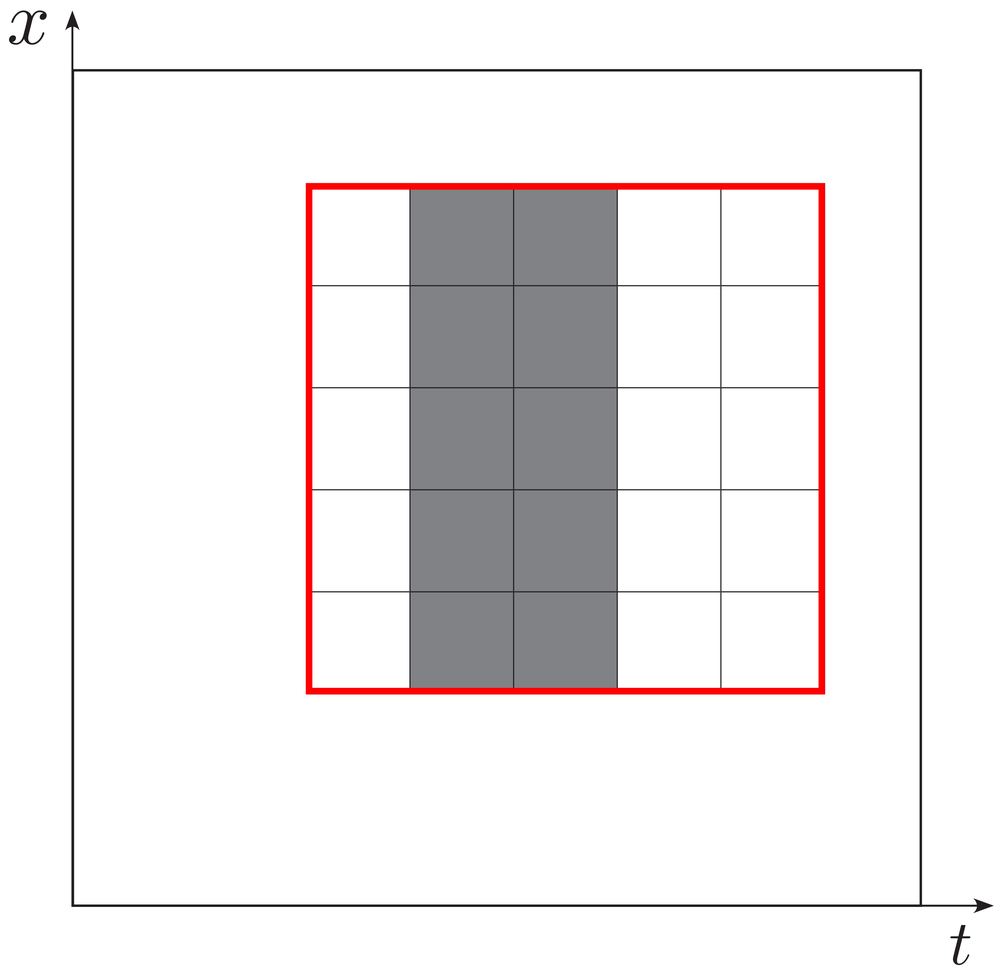
Zero-shot capabilities



Zero-shot capabilities

Also:

- spatial interpolation (missing data)
- downscaling
- ...



Training task:

predict randomly masked neighborhoods in space-time

Model correction

- Numerical statistical atmospheric model:

$$p_\theta\left(y|x,\alpha\right)$$

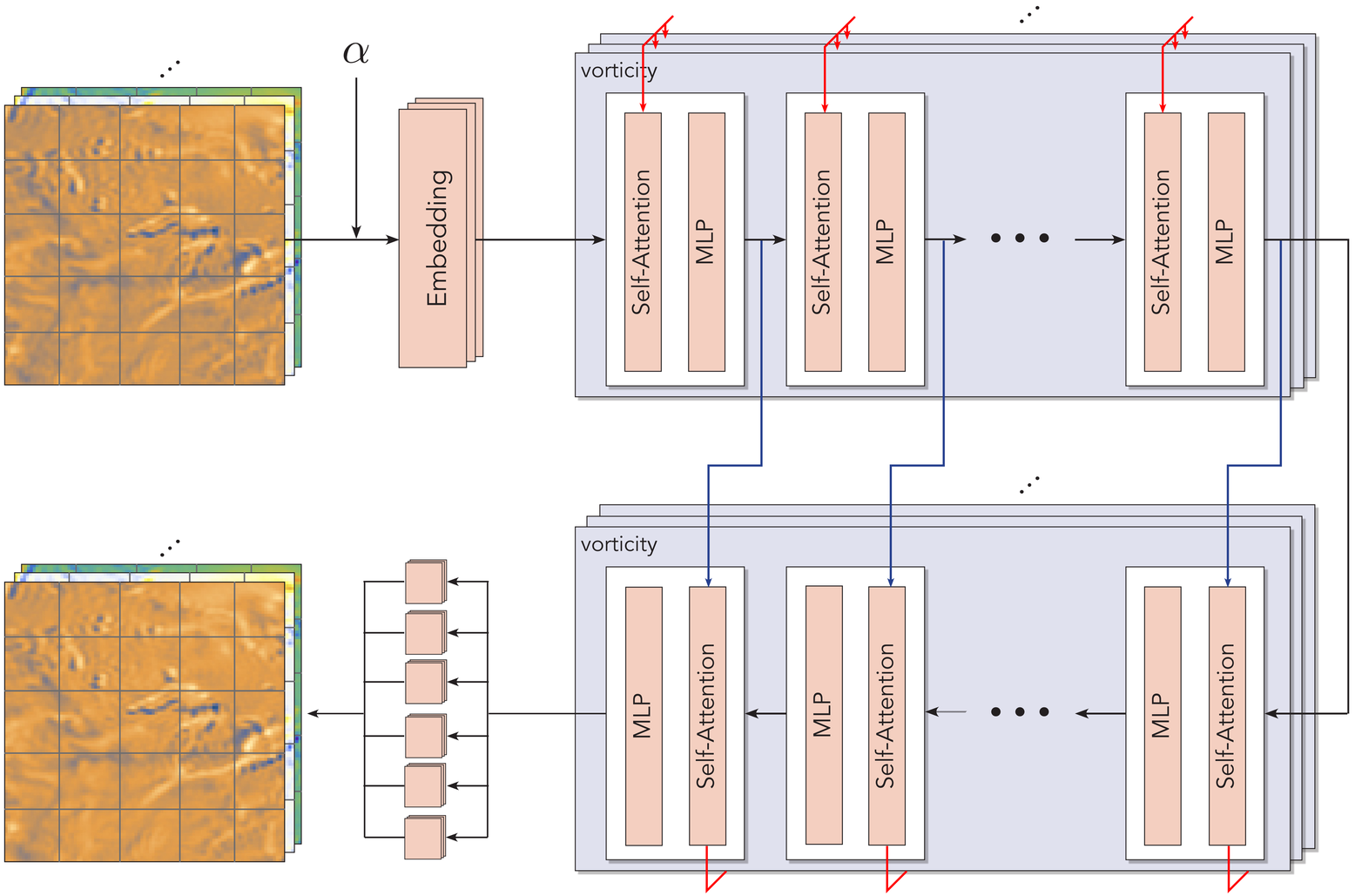
Model correction

- Numerical statistical atmospheric model:

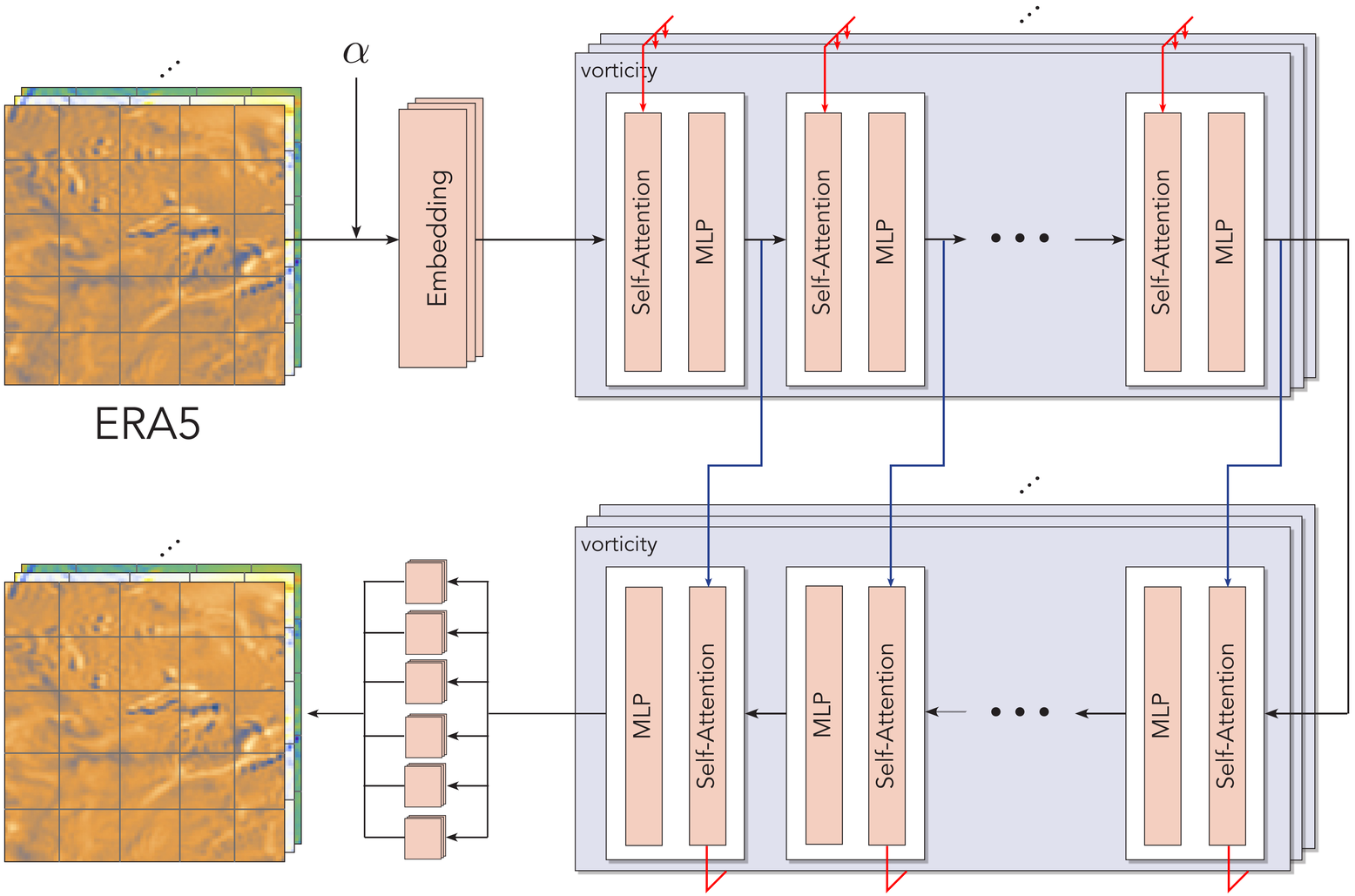
$$p_\theta\left(y|x,\alpha\right)$$

approximate initial state

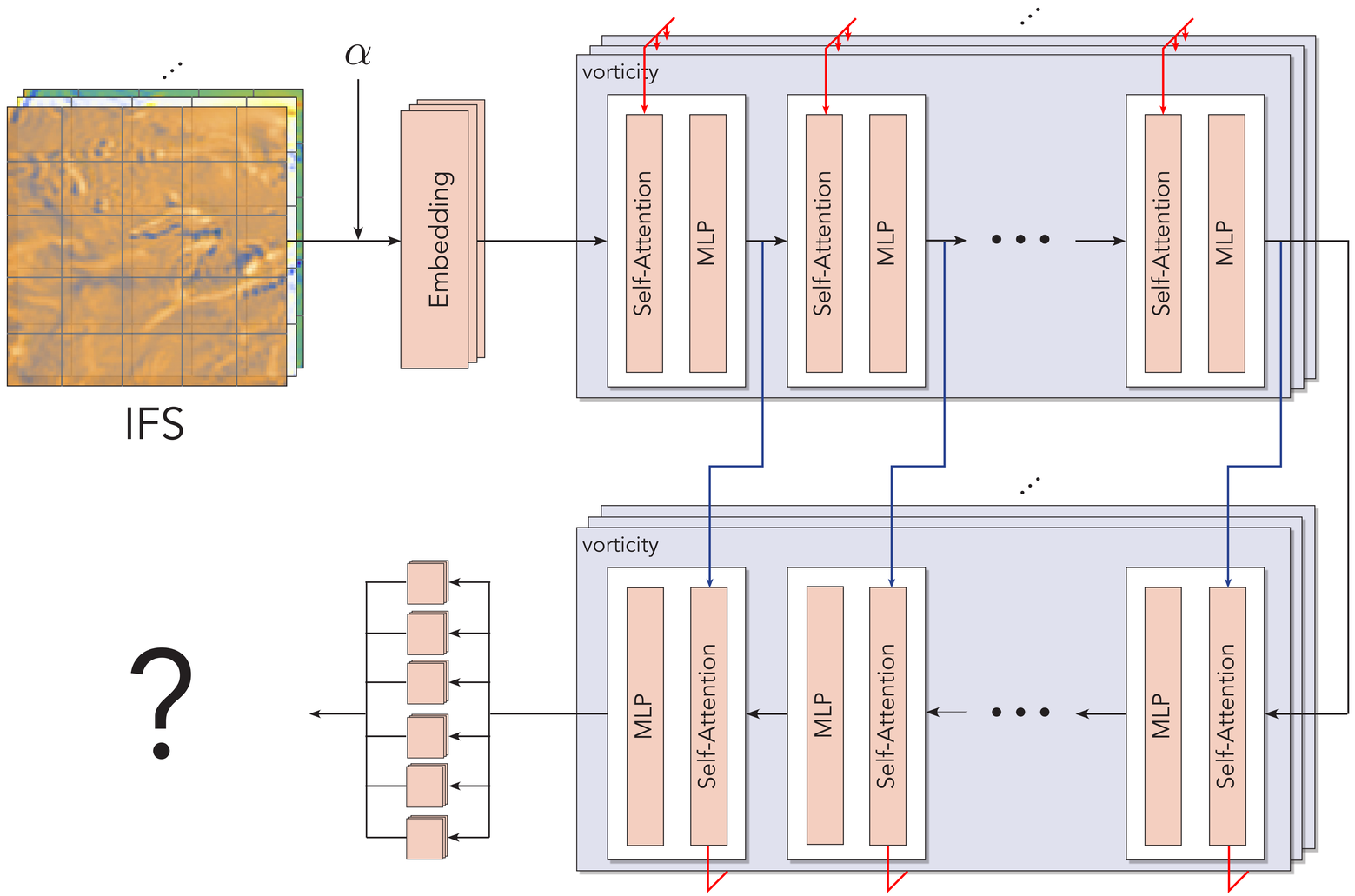
Model correction



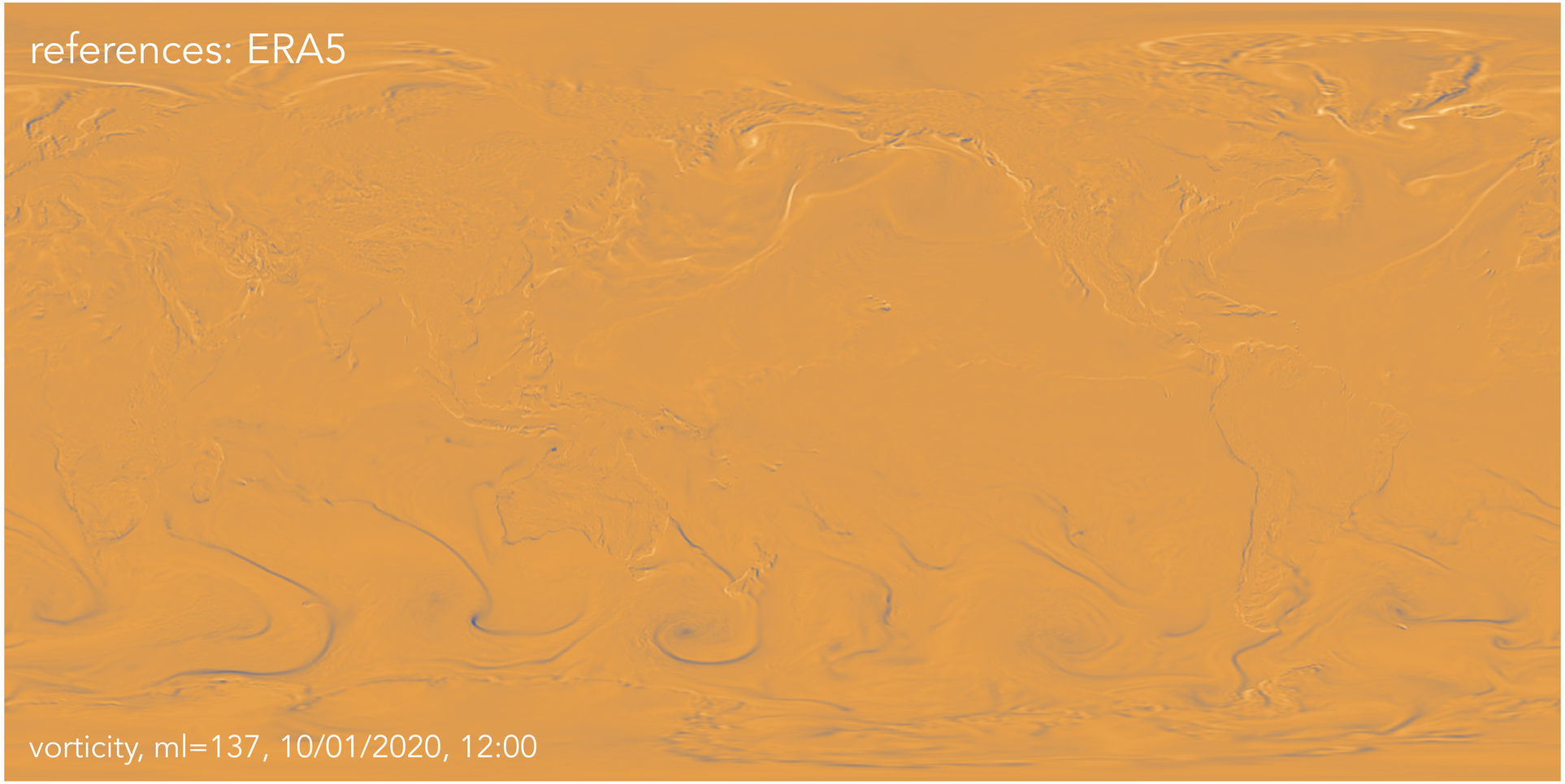
Model correction



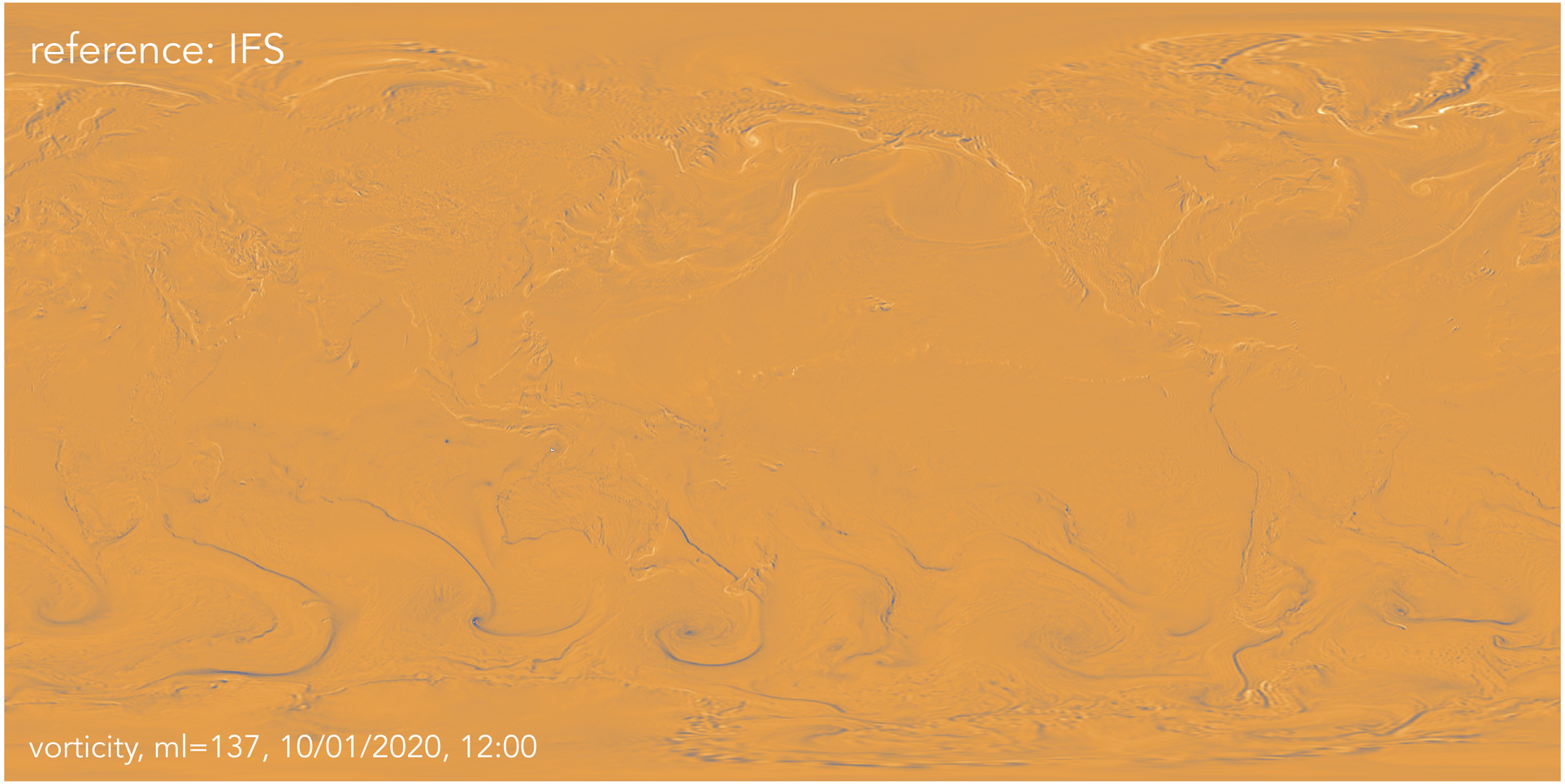
Model correction



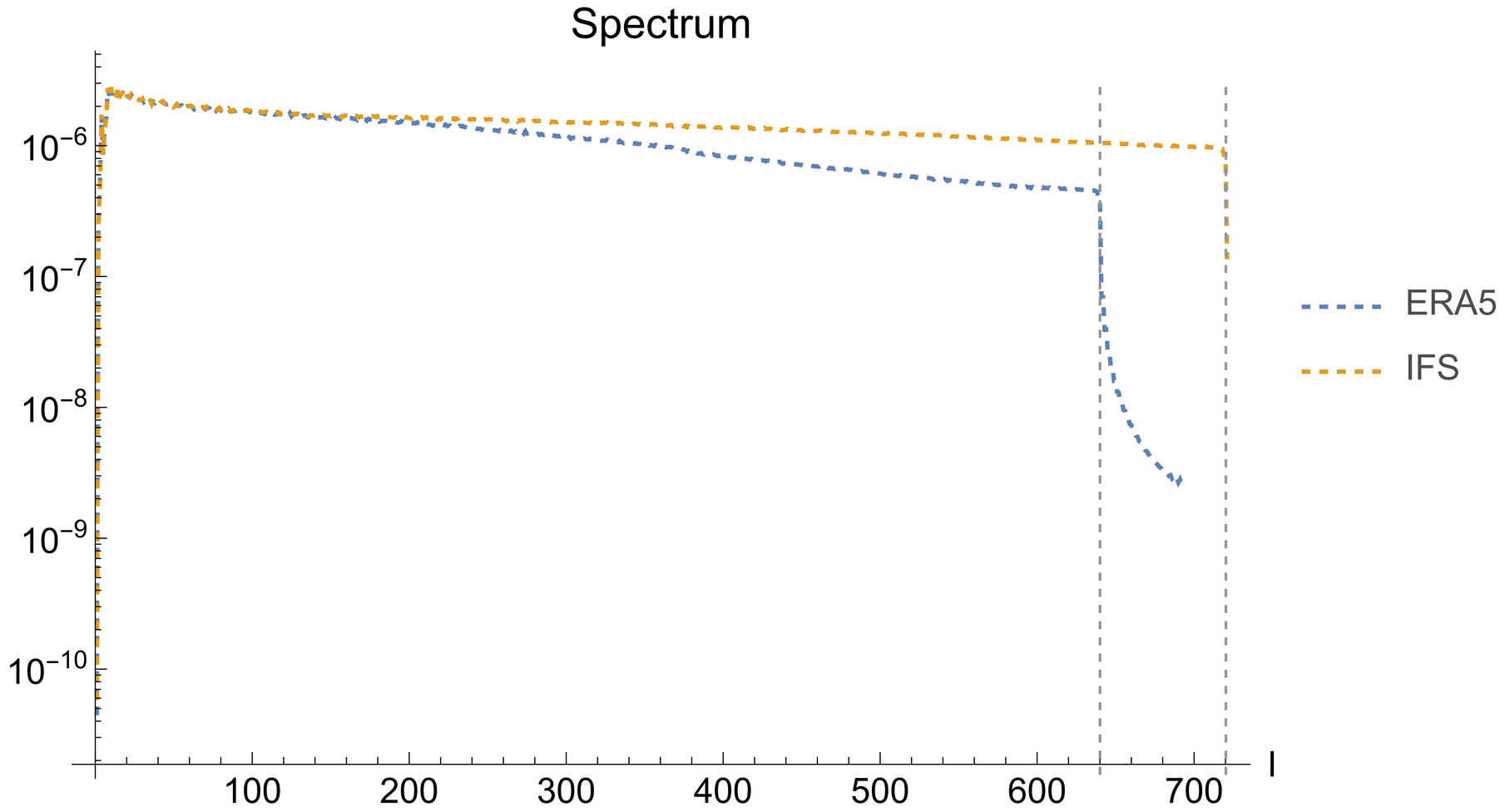
Model correction



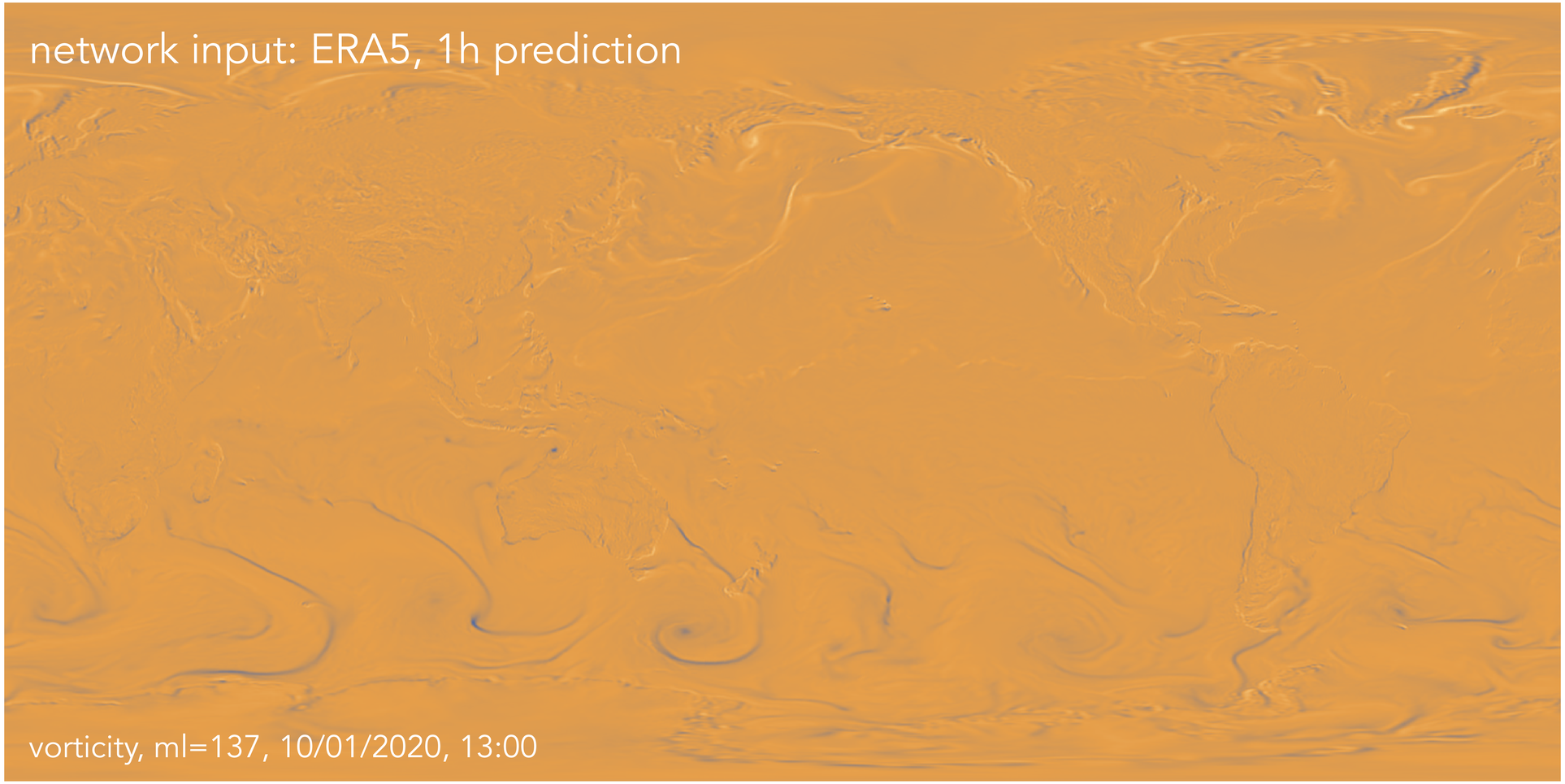
Model correction



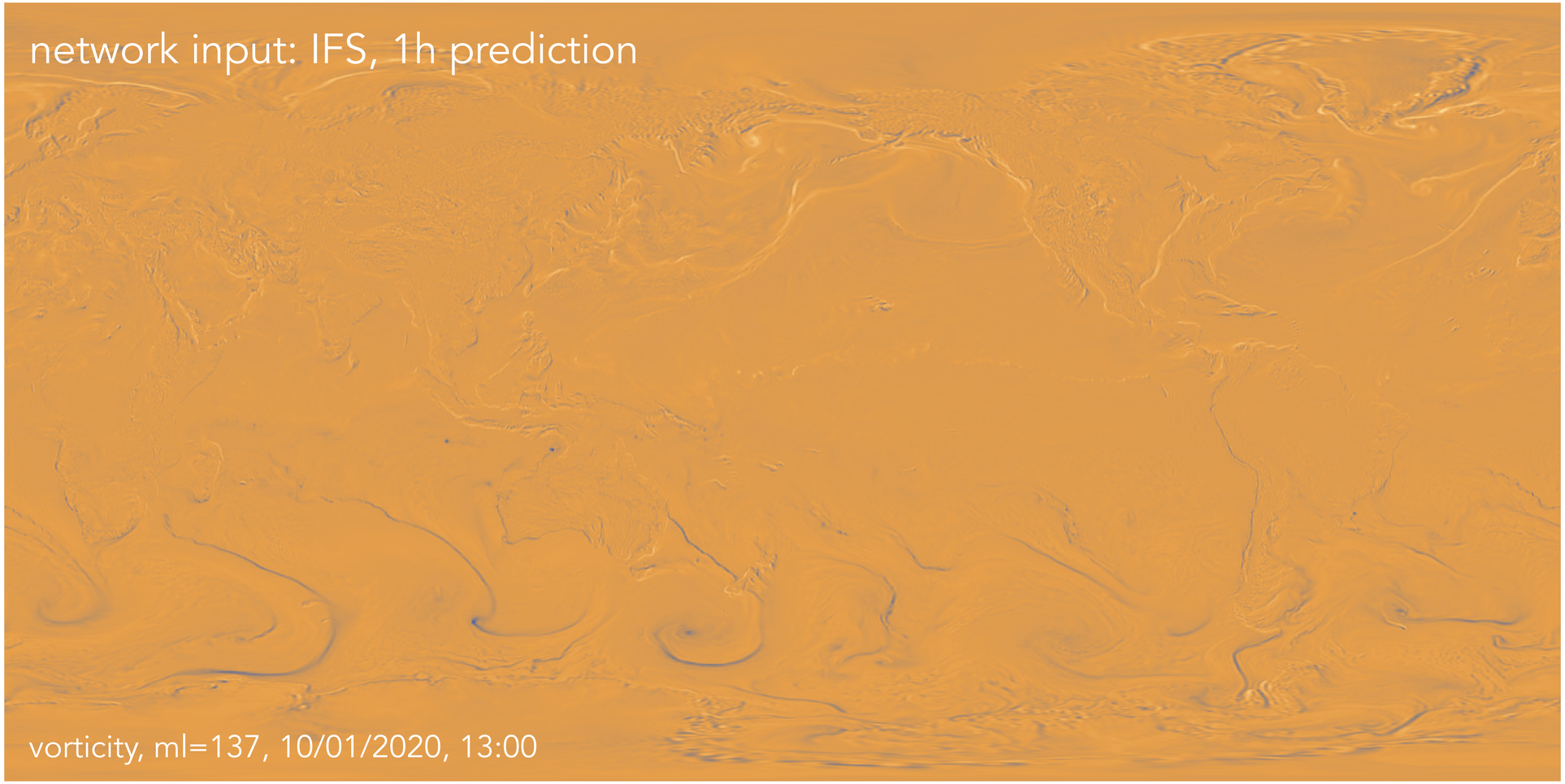
Model correction



Model correction



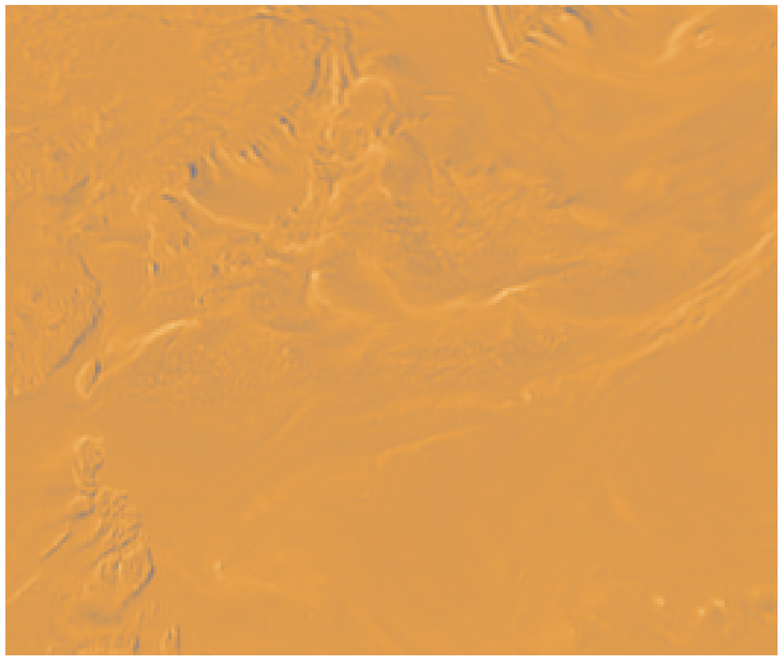
Model correction



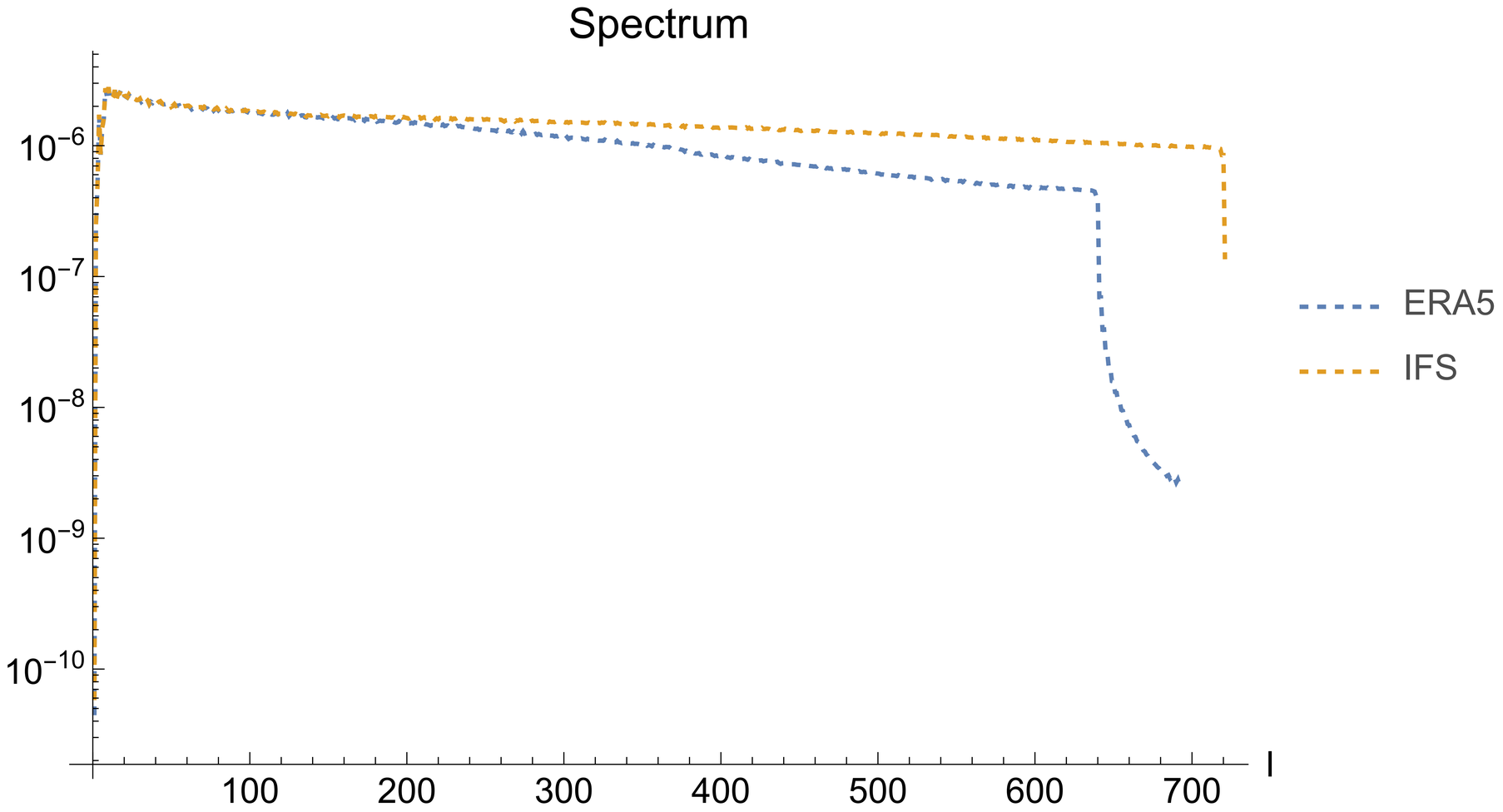
Model correction

network input: ERA5

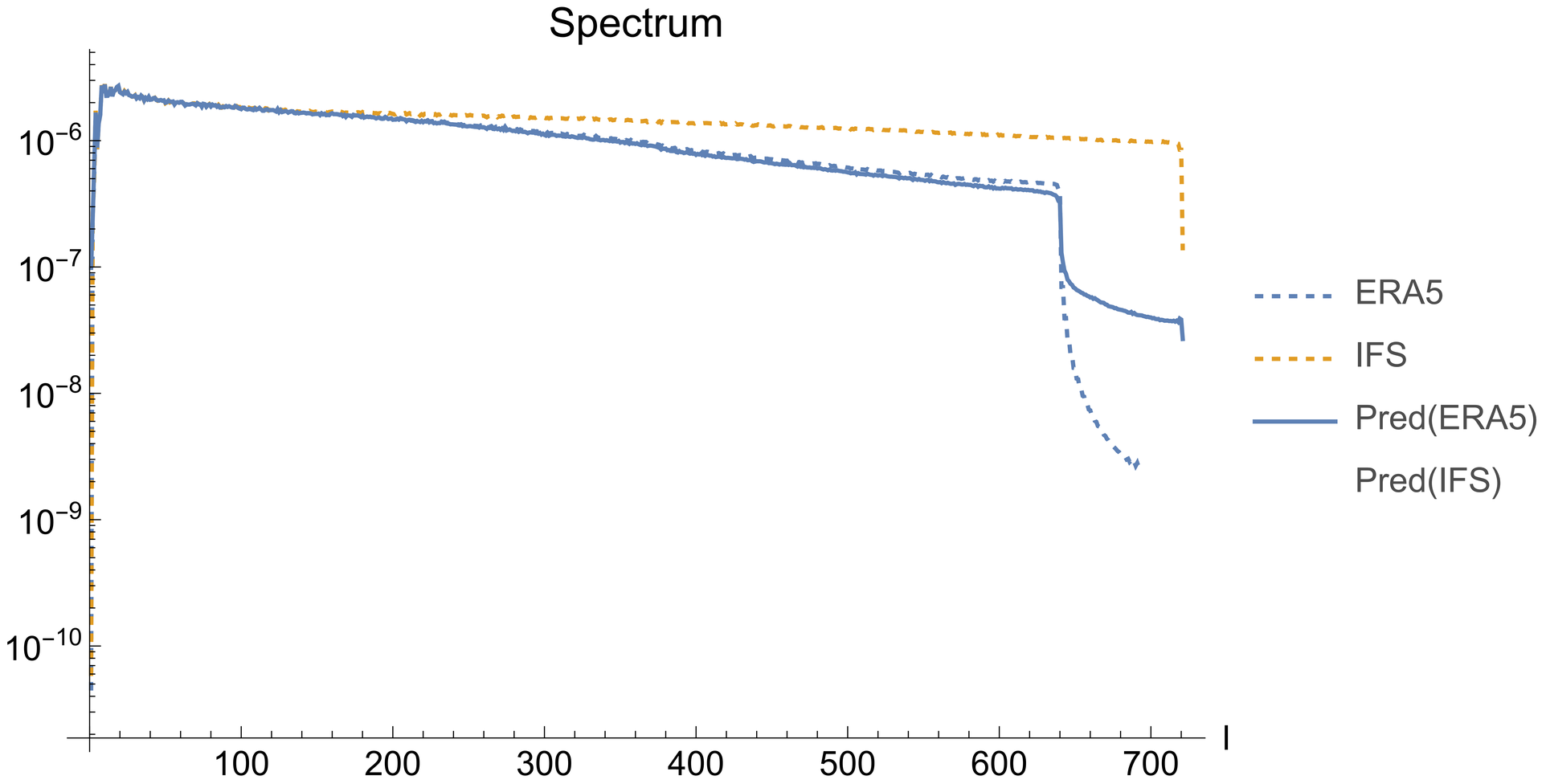
network input: IFS



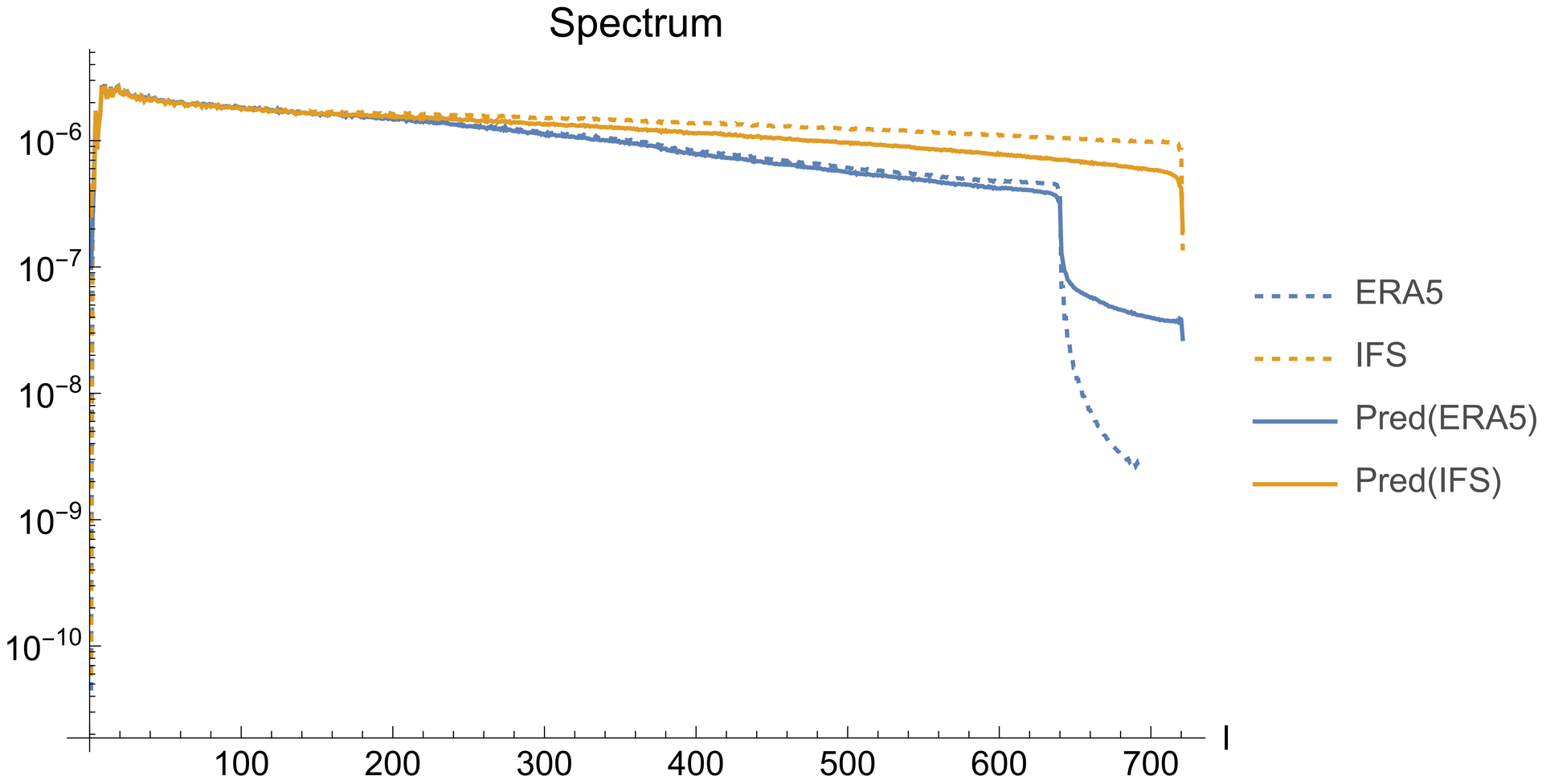
Model correction



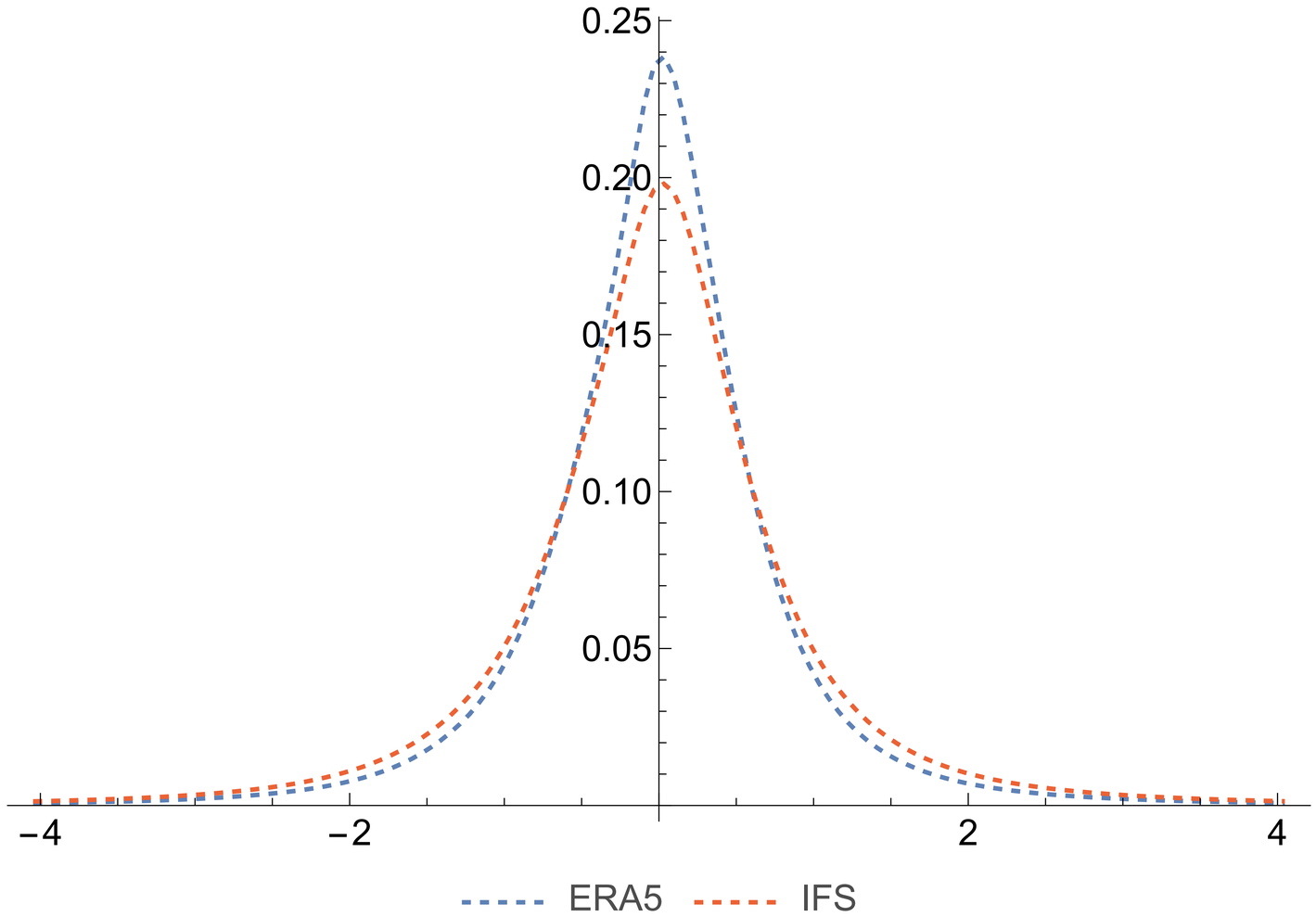
Model correction



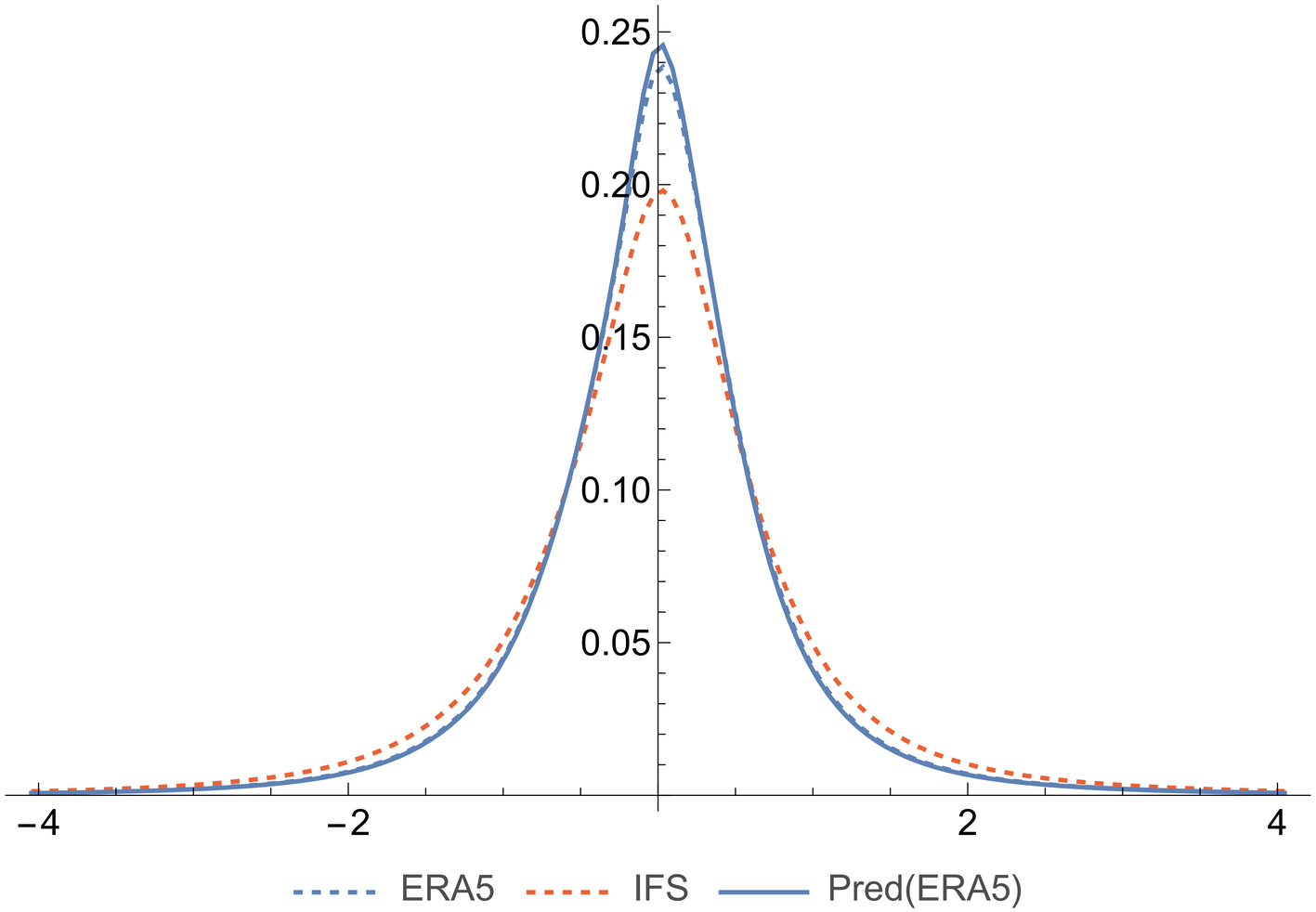
Model correction



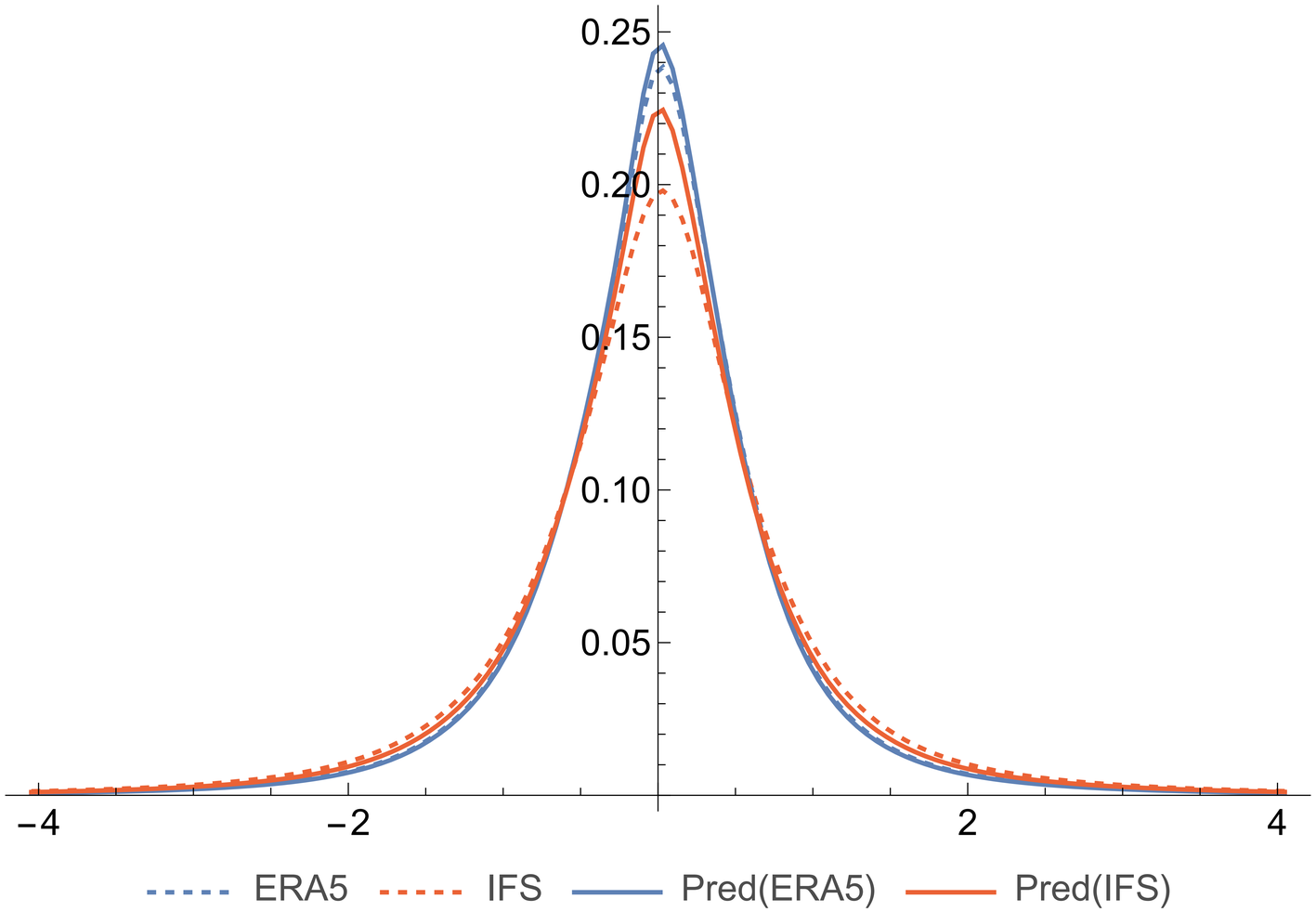
Model correction



Model correction



Model correction



Counterfactuals

- Numerical statistical atmospheric model:

$$p_\theta\left(y|x,\alpha\right)$$

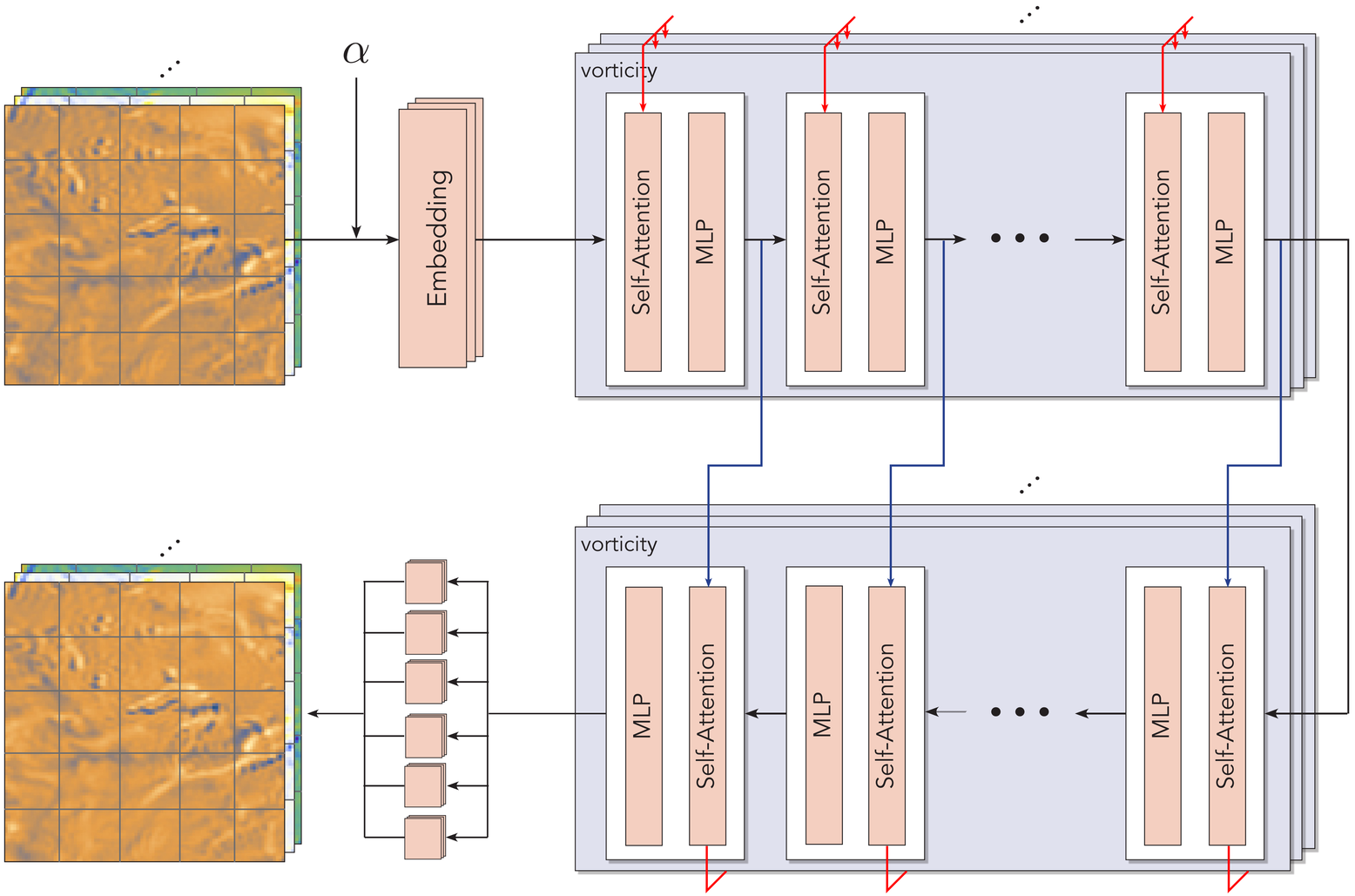
Counterfactuals

- Numerical statistical atmospheric model:

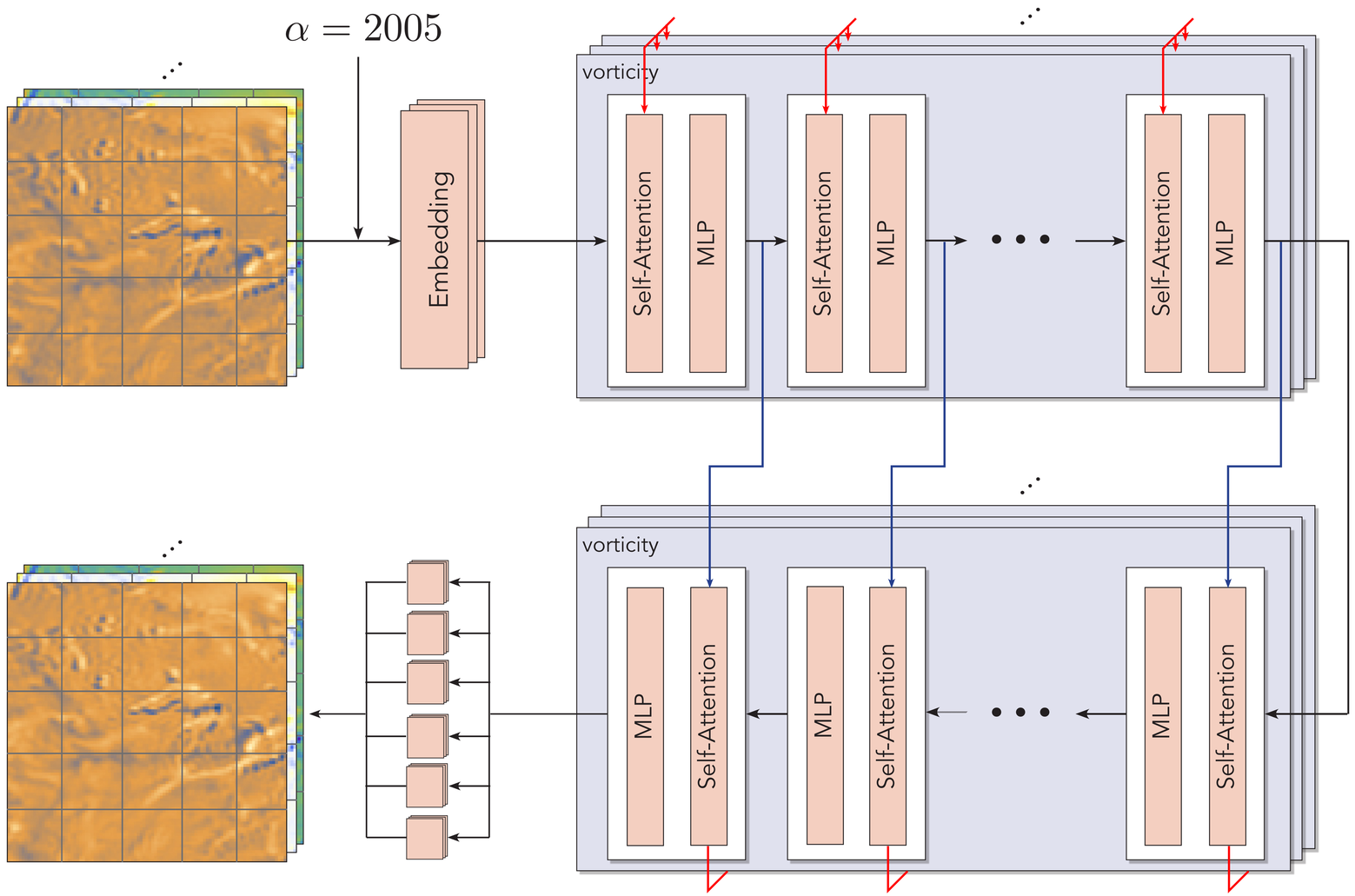
$$p_\theta\left(y|x,\alpha\right)$$

$$\alpha = \left(\mathrm{year}, \mathrm{day}, \mathrm{hour}, \mathrm{ml}, \theta, \phi, \mathrm{res}\right)$$

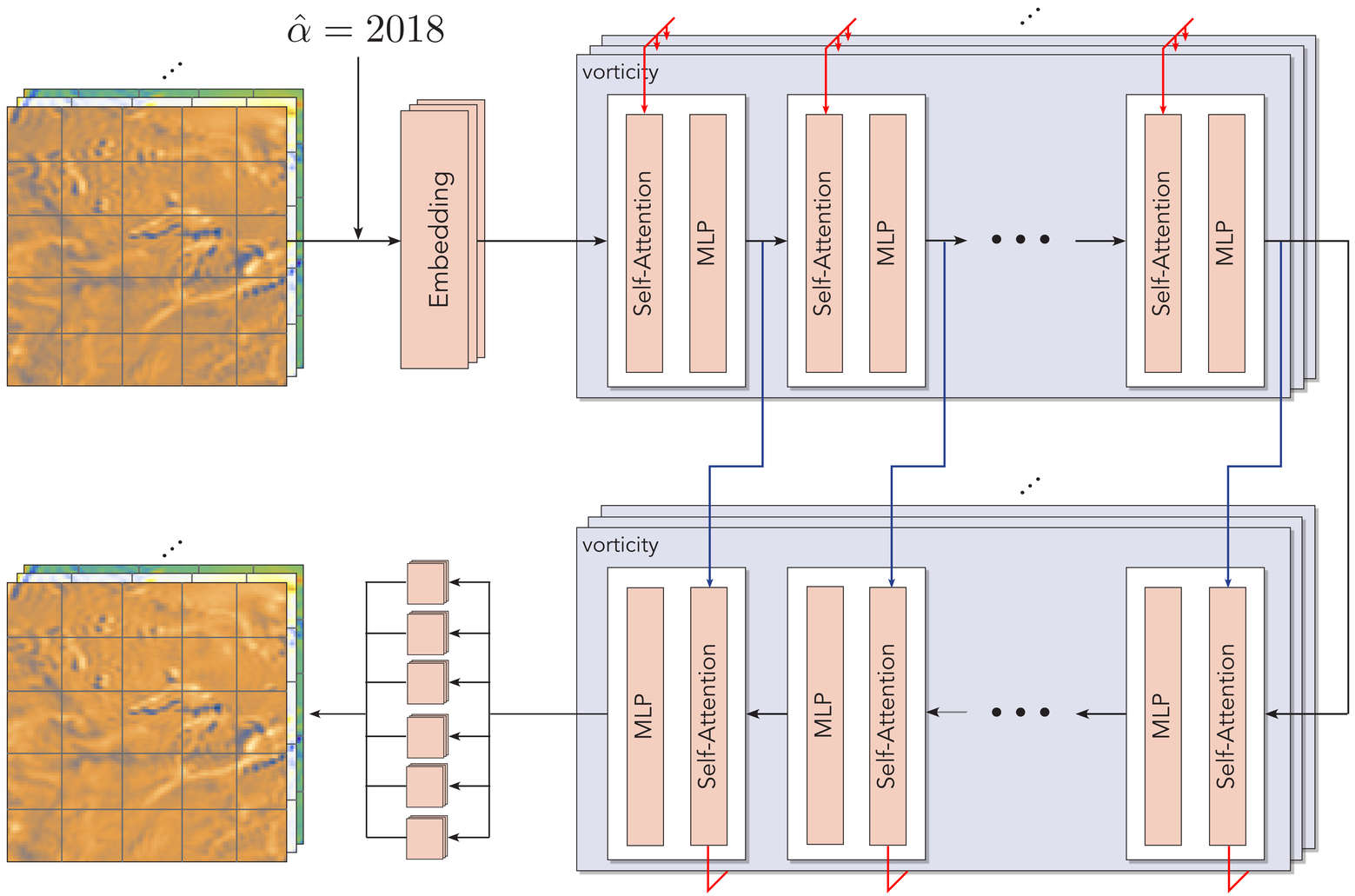
Counterfactuals



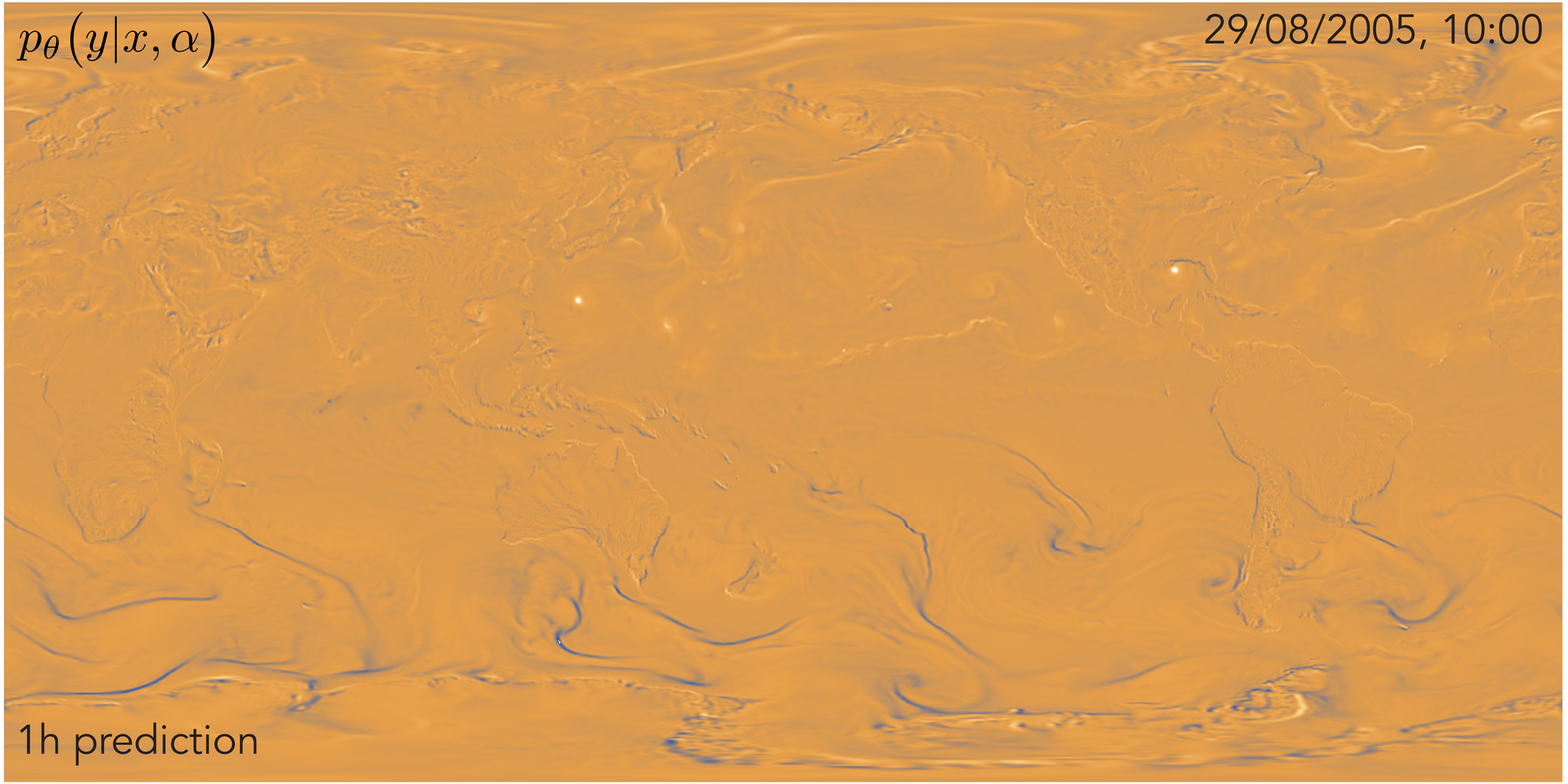
Counterfactuals



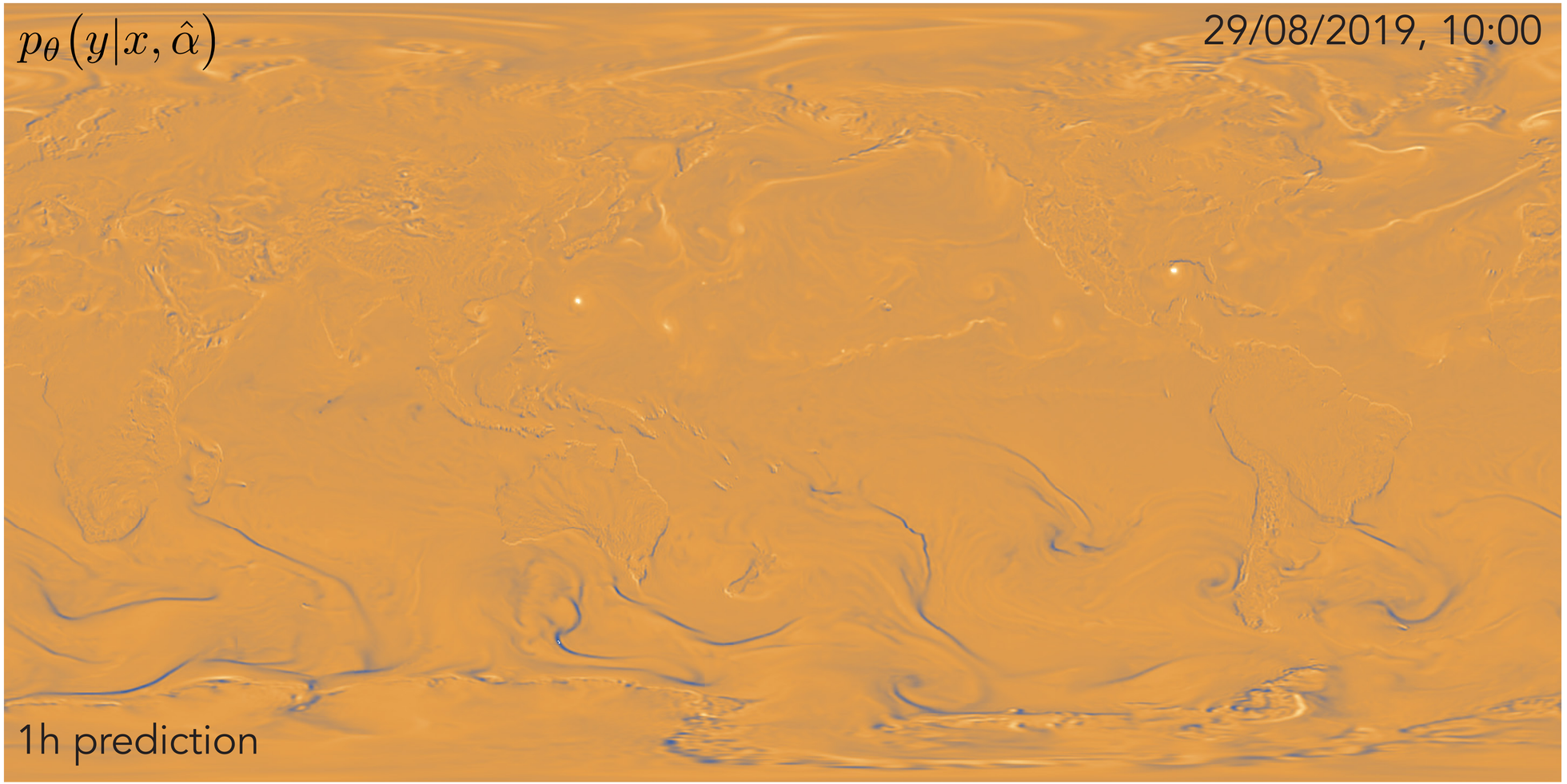
Counterfactuals



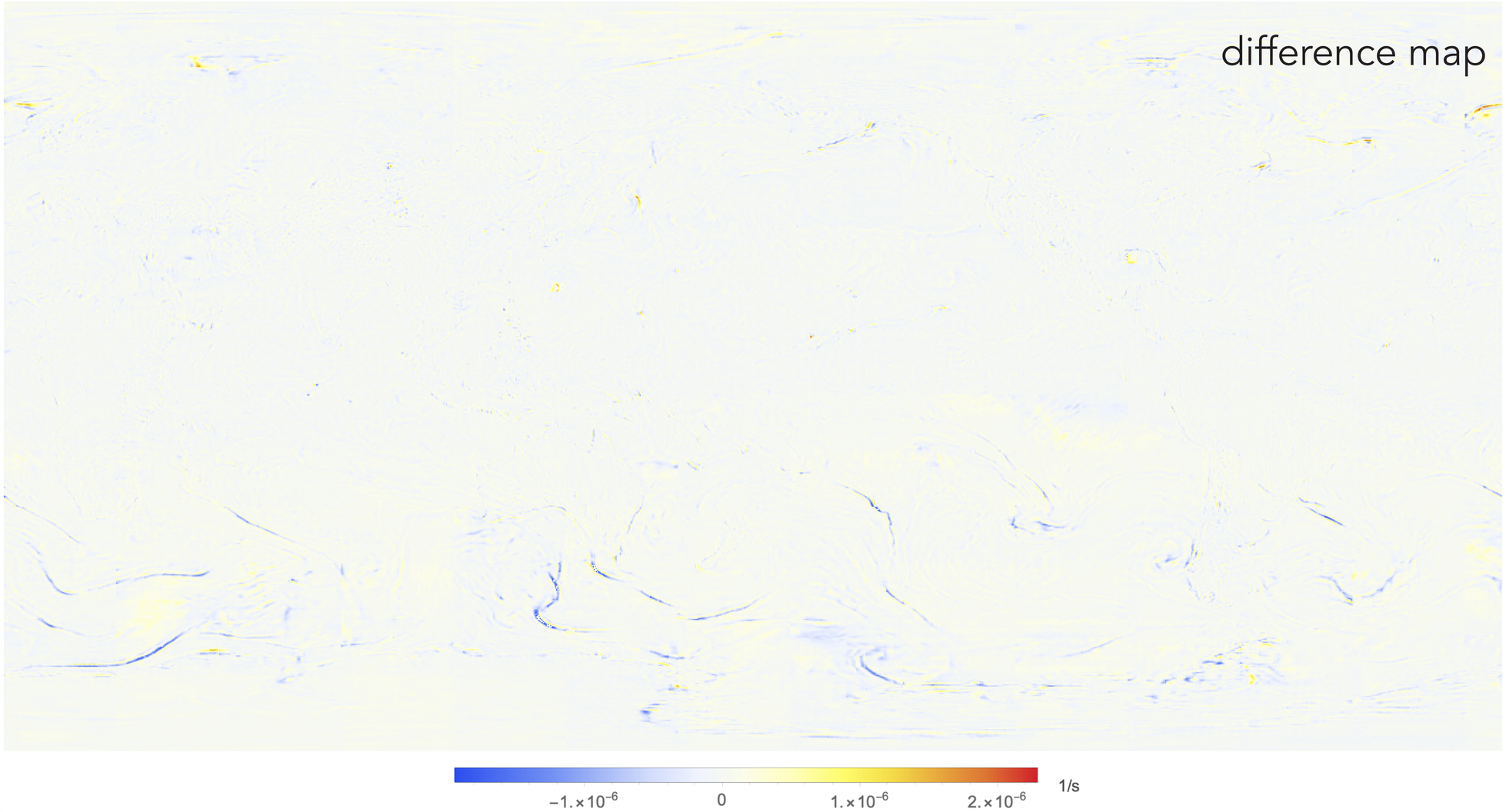
Counterfactuals



Counterfactuals



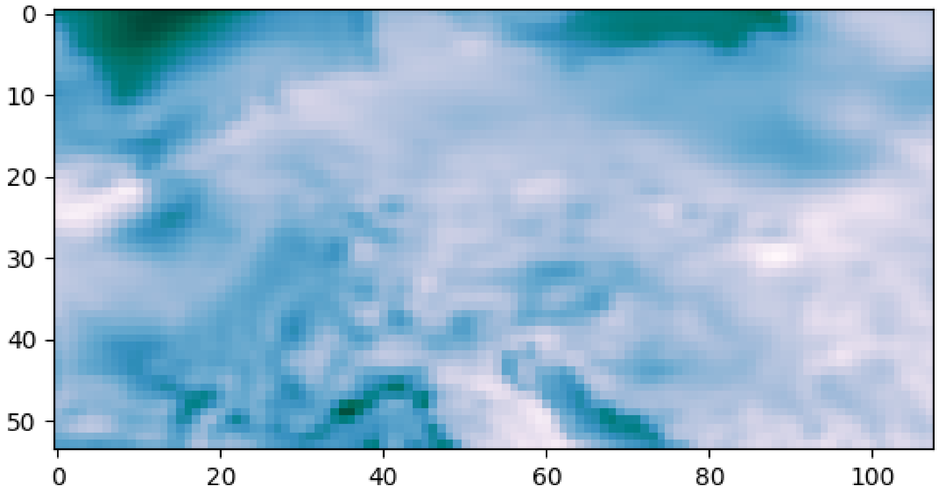
Counterfactuals



Downstream applications

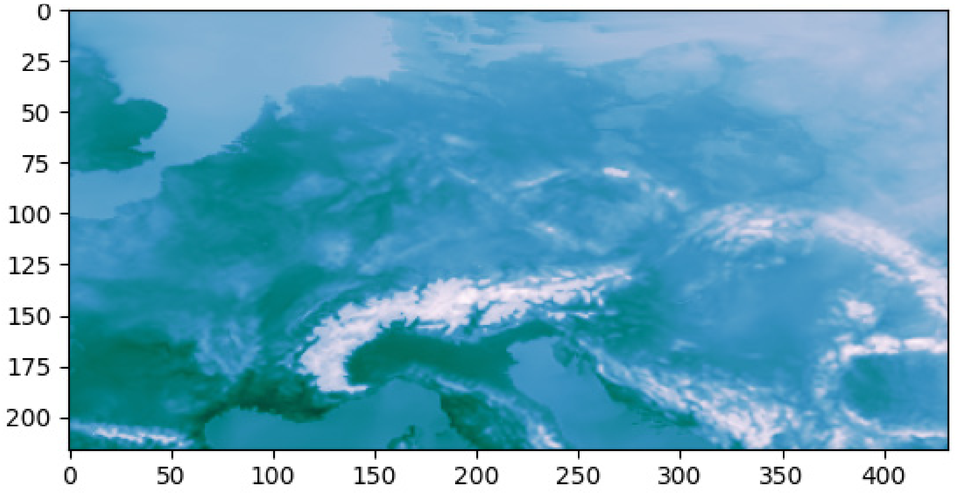
Downscaling

ERA5



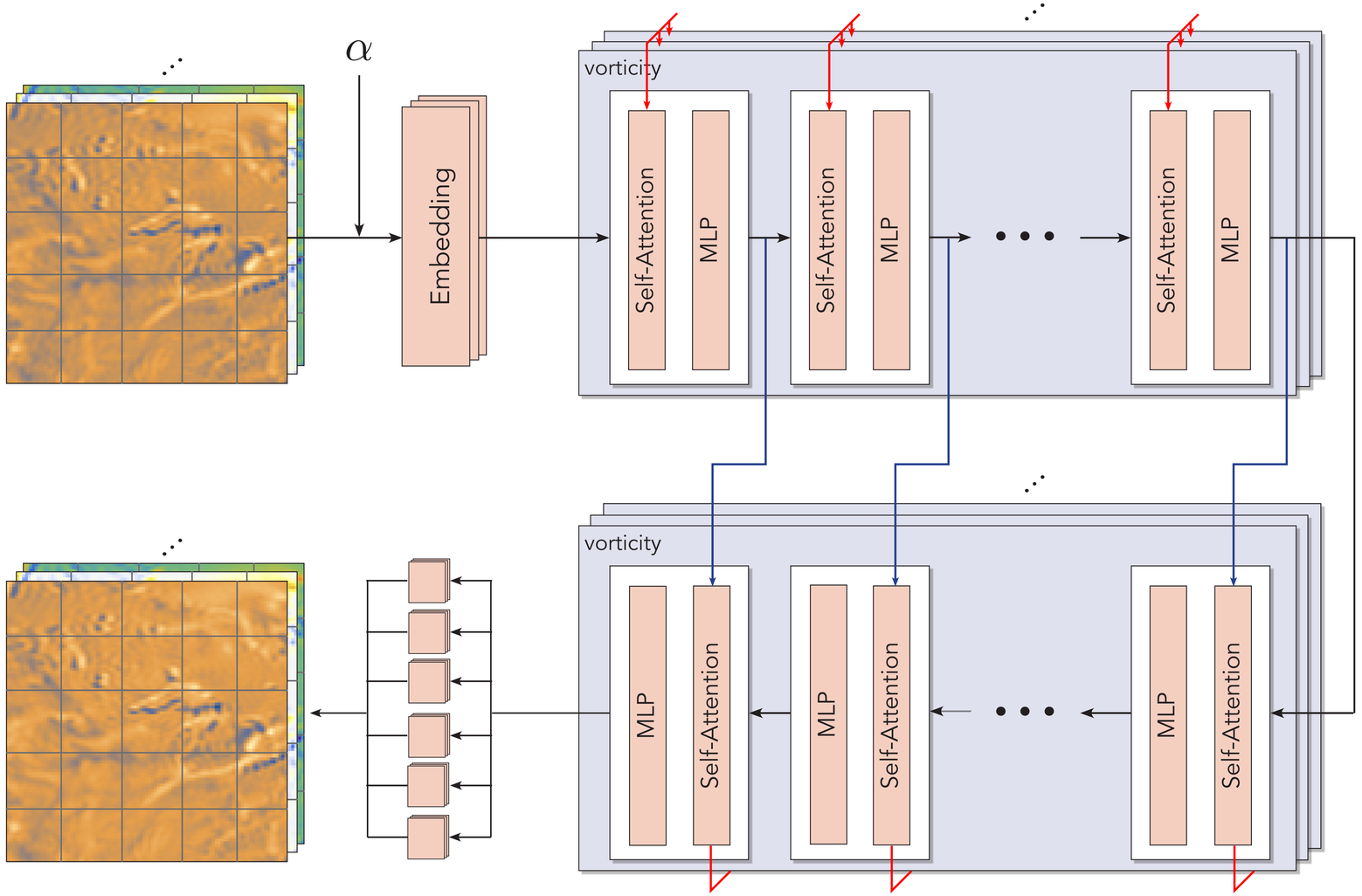
temperature, ml=137

COSMO-REA6

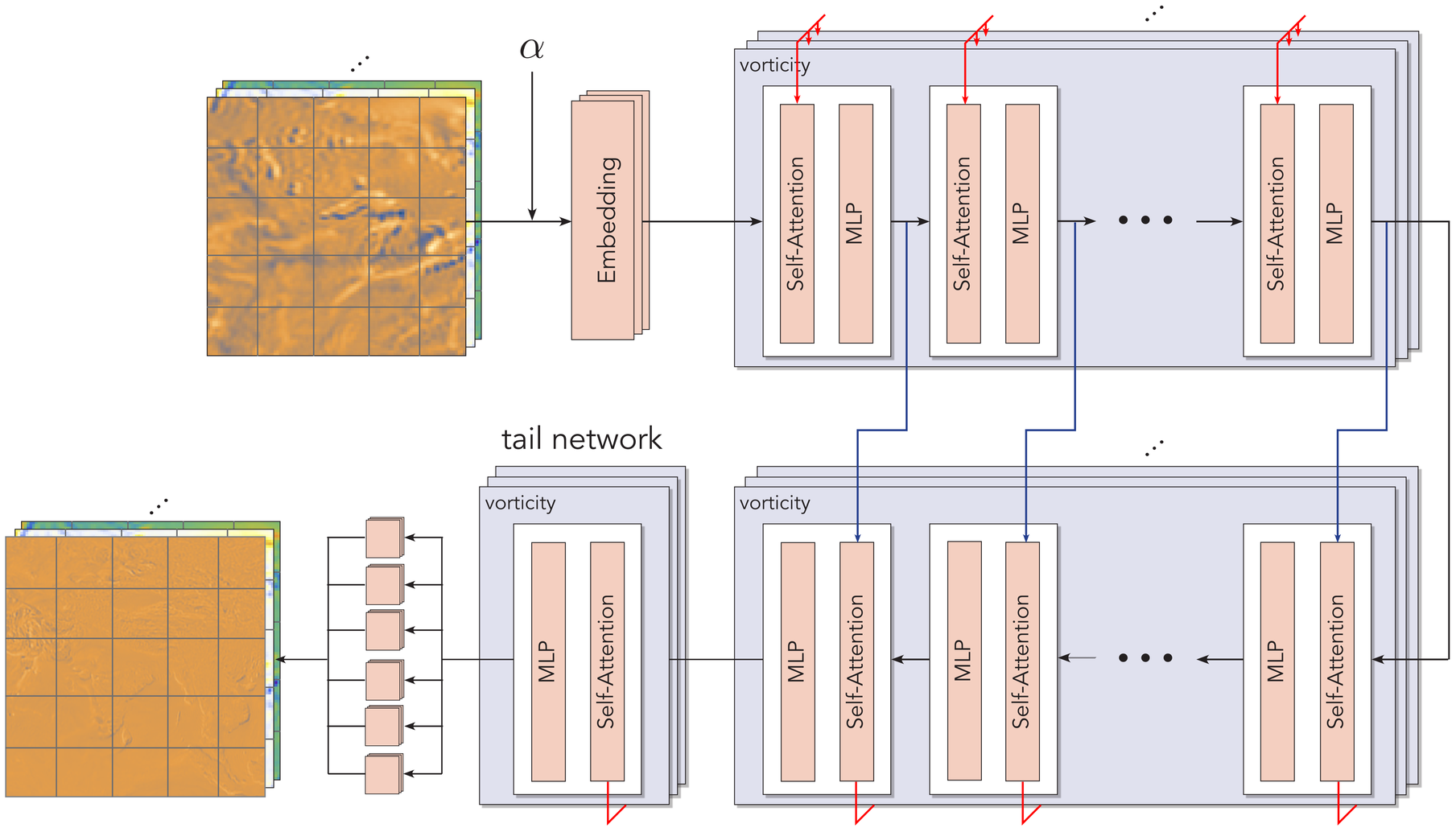


temperature, 2 m

Downscaling



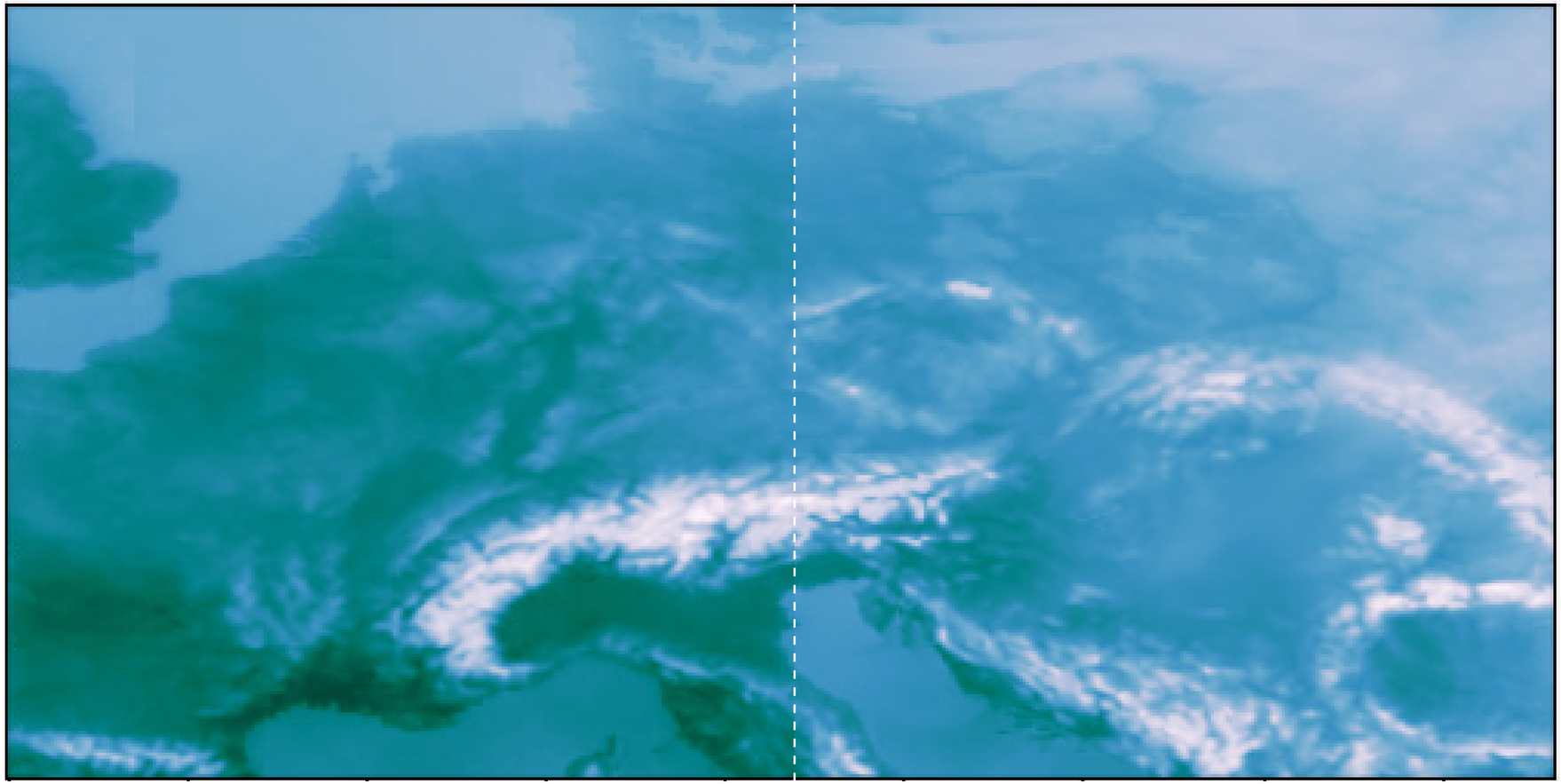
Downscaling



Downscaling

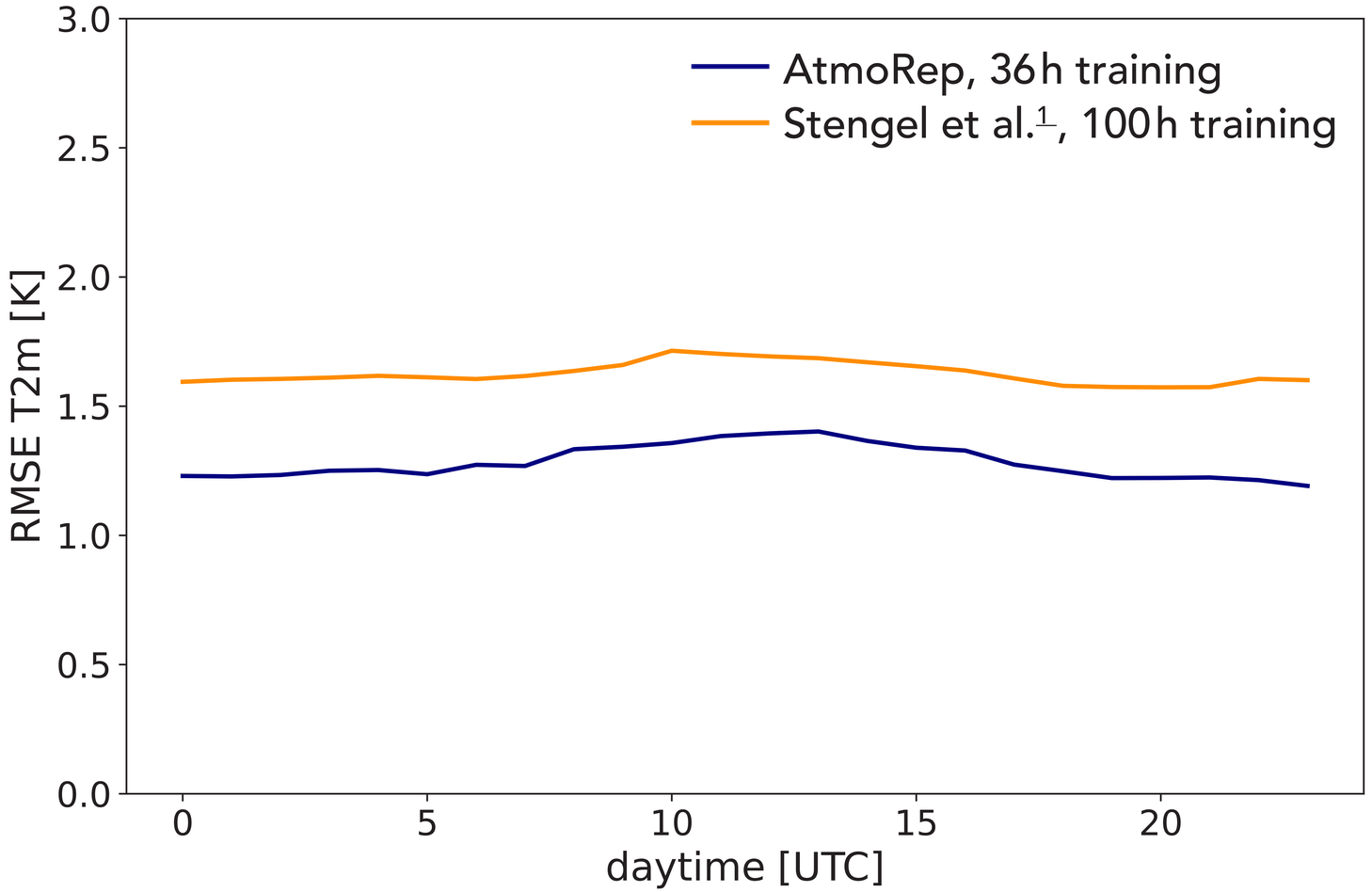
AtmoRep prediction

COSMO-REA6



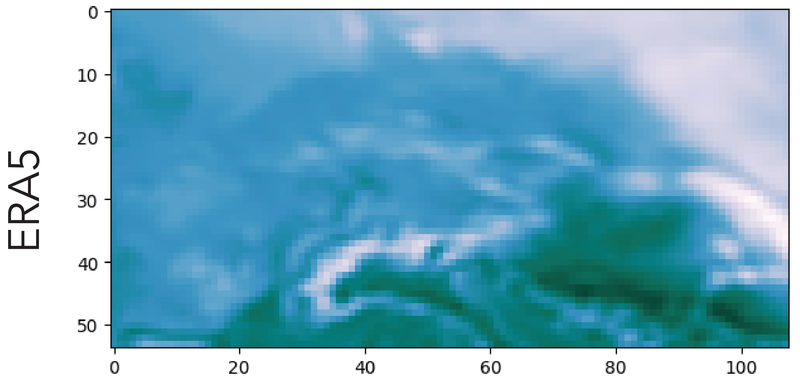
temperature, 2 m

Downscaling

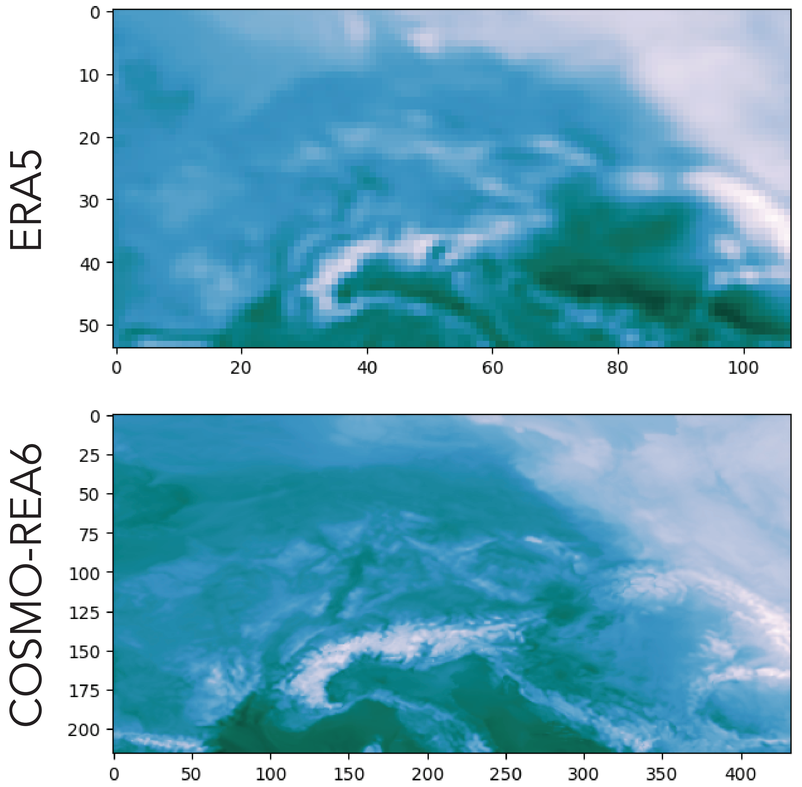


[1] K. Stengel, A. Glaws, D. Hettinger, and R. N. King. Adversarial super-resolution of climatological wind and solar data. Proceedings of the National Academy of Sciences, 117(29):16805–16815, 2020.

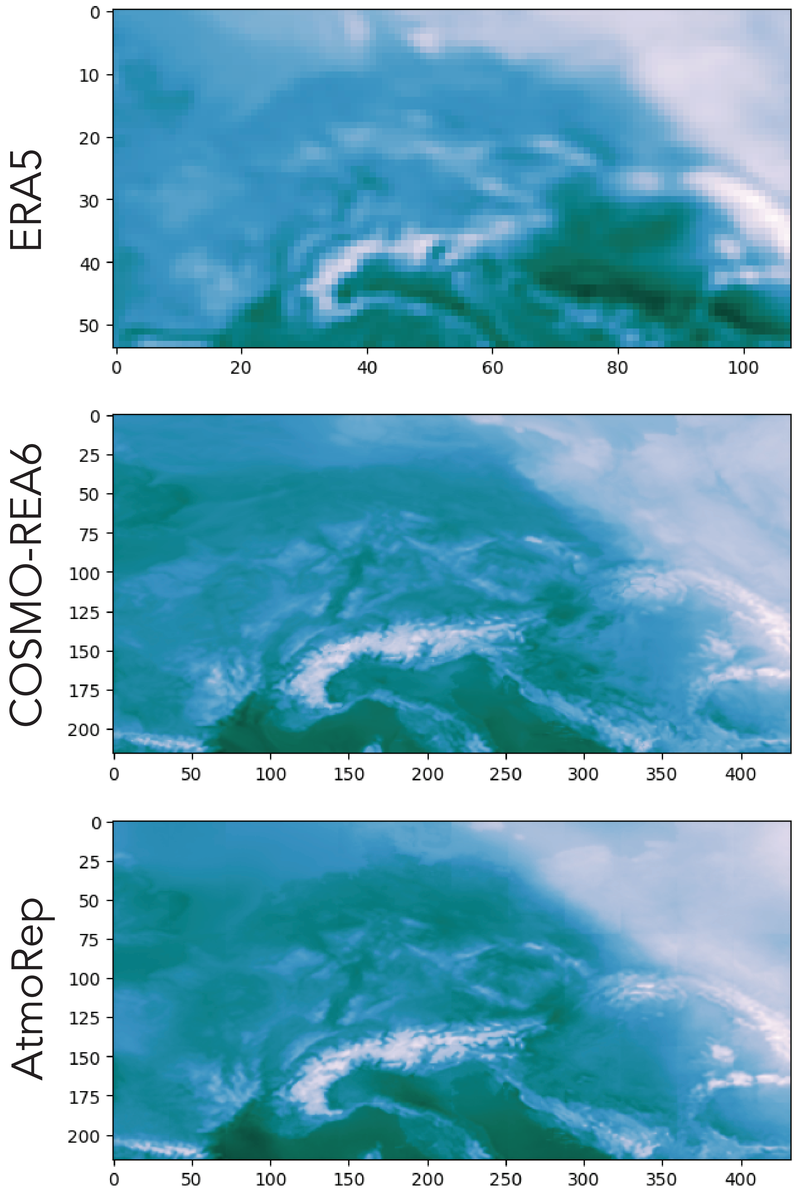
Downscaling



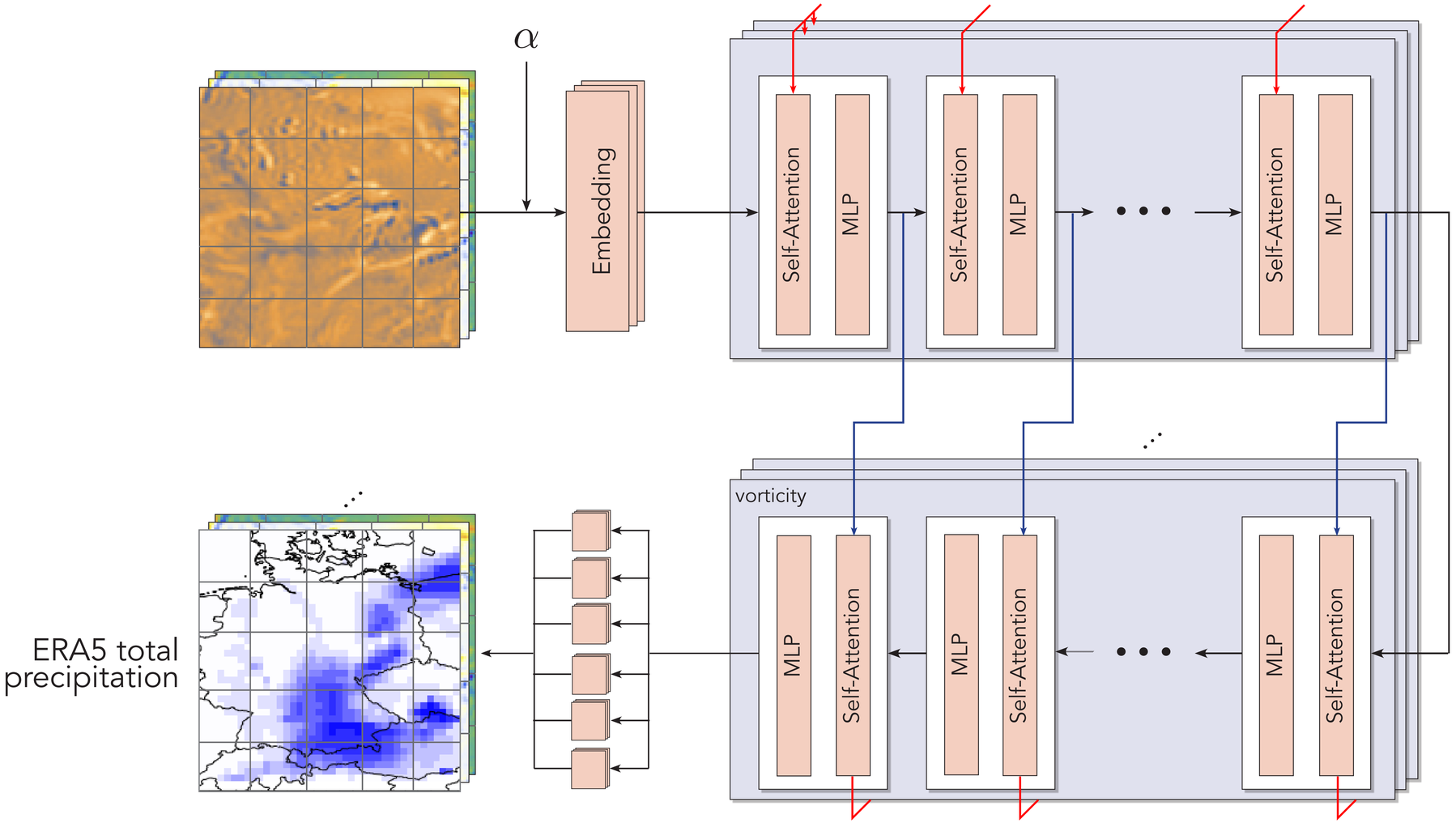
Downscaling



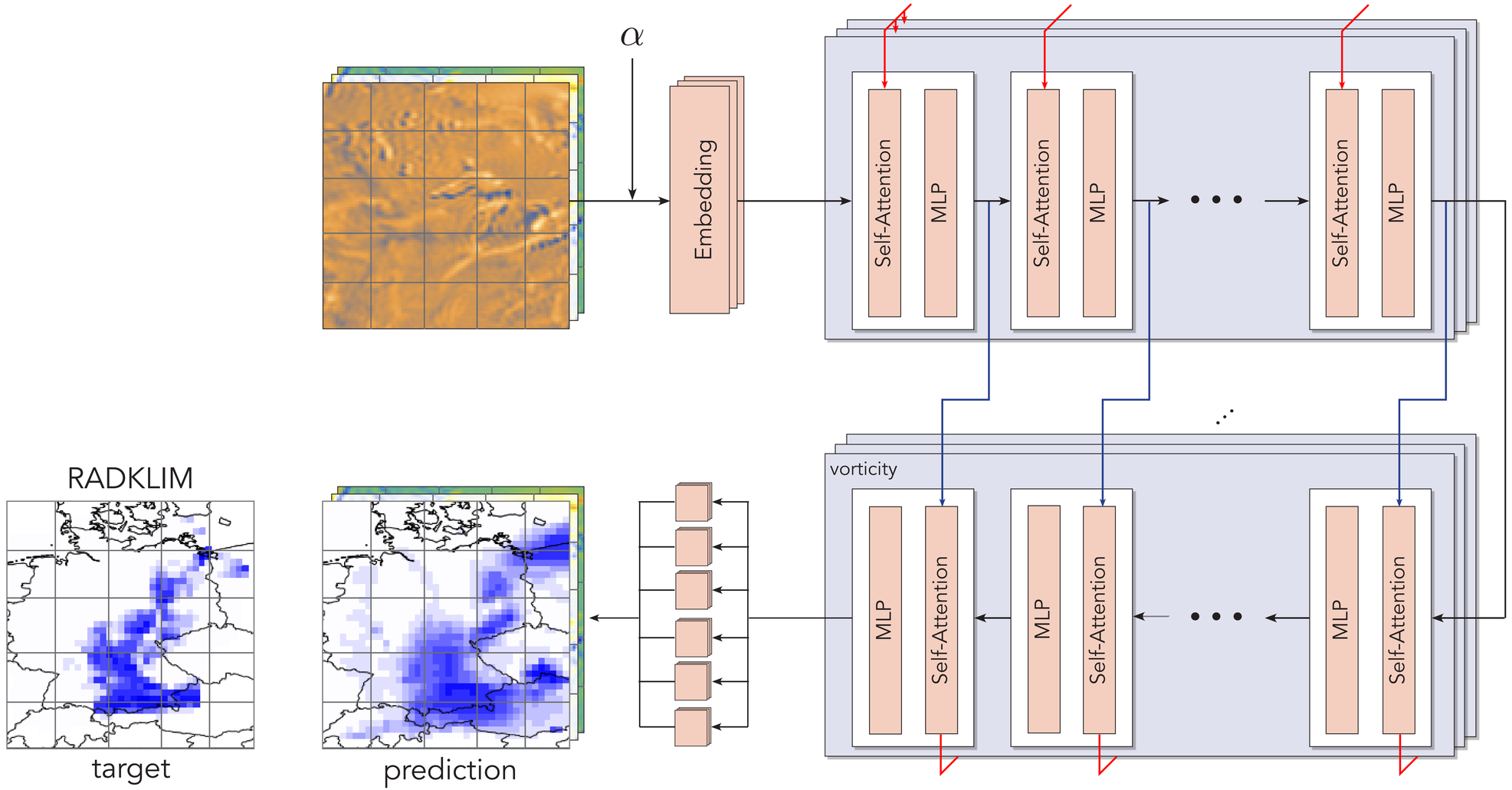
Downscaling



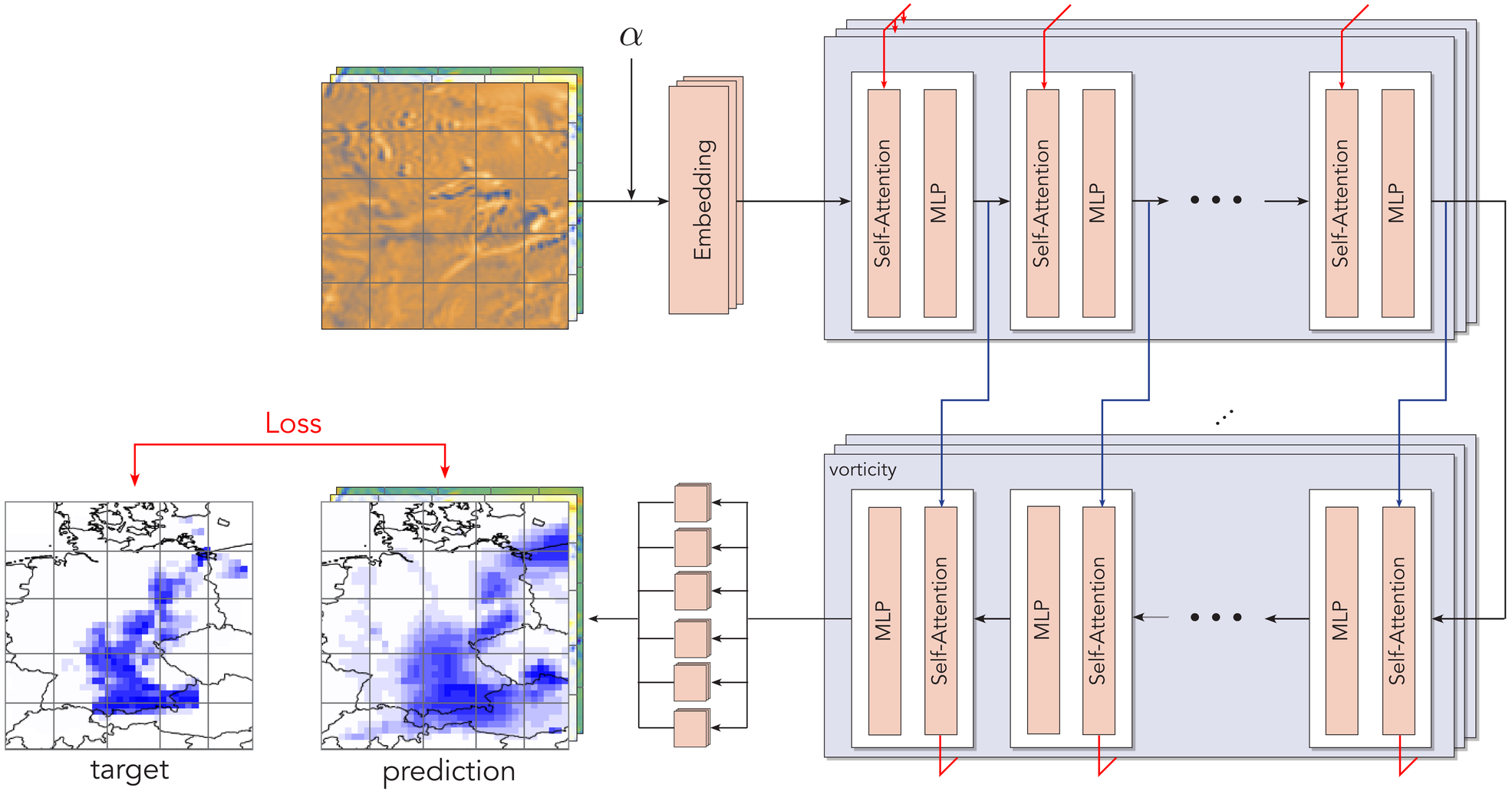
Bias correction



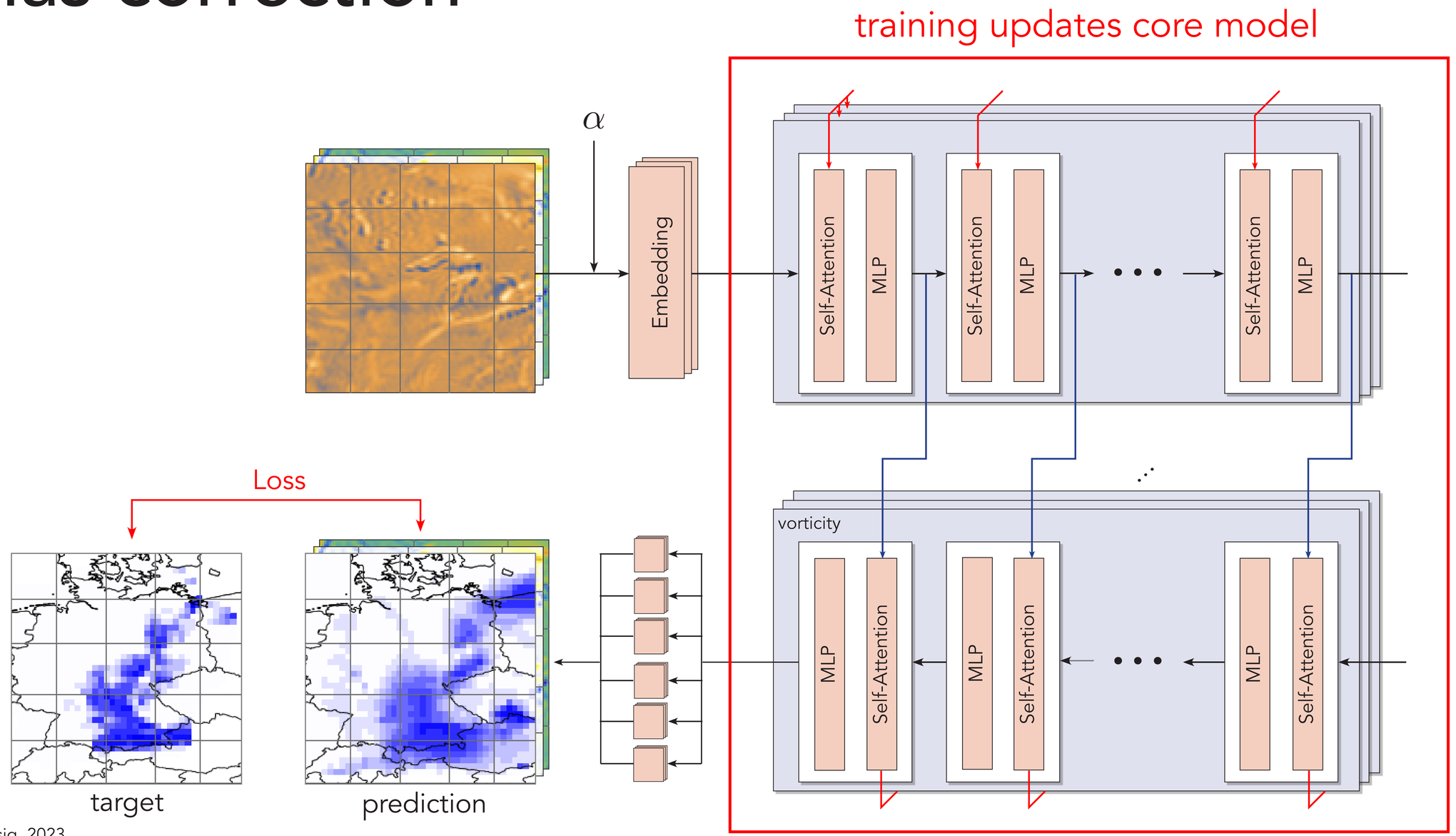
Bias correction



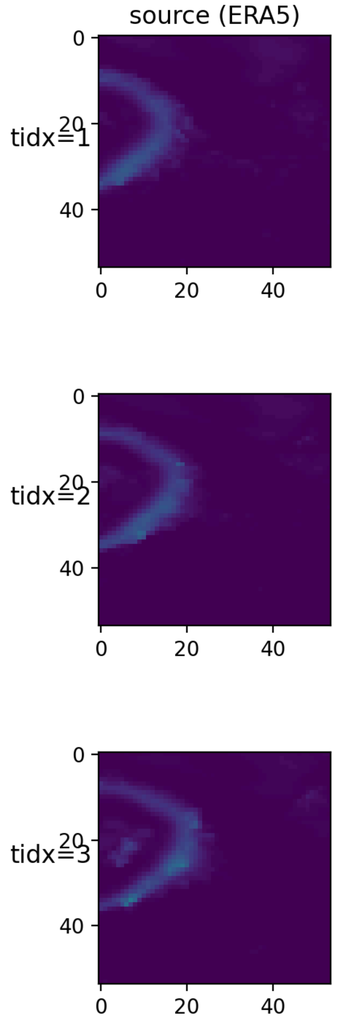
Bias correction



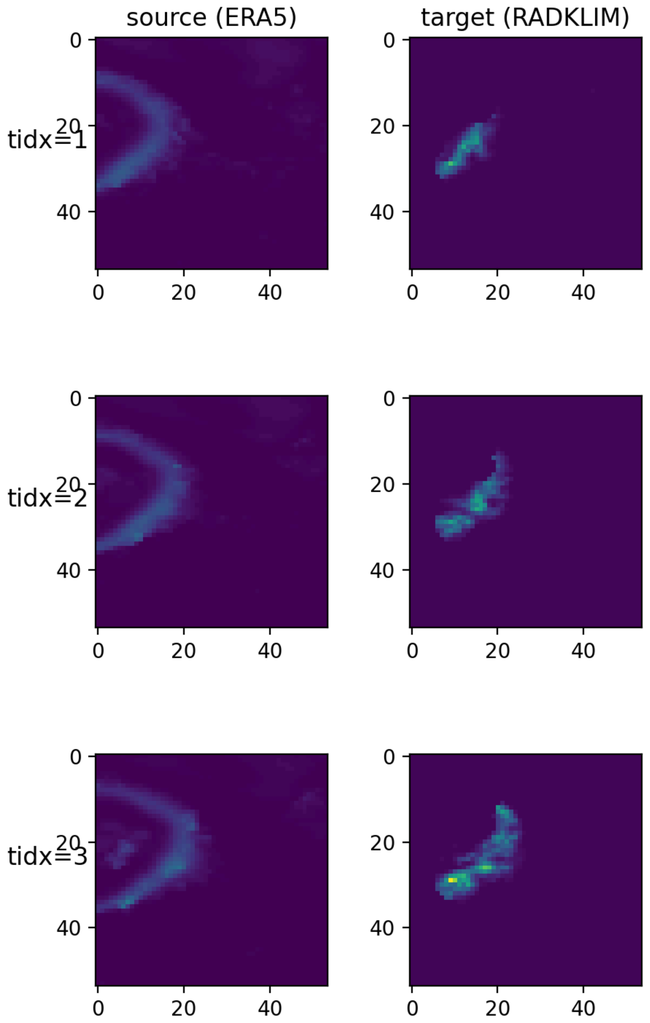
Bias correction



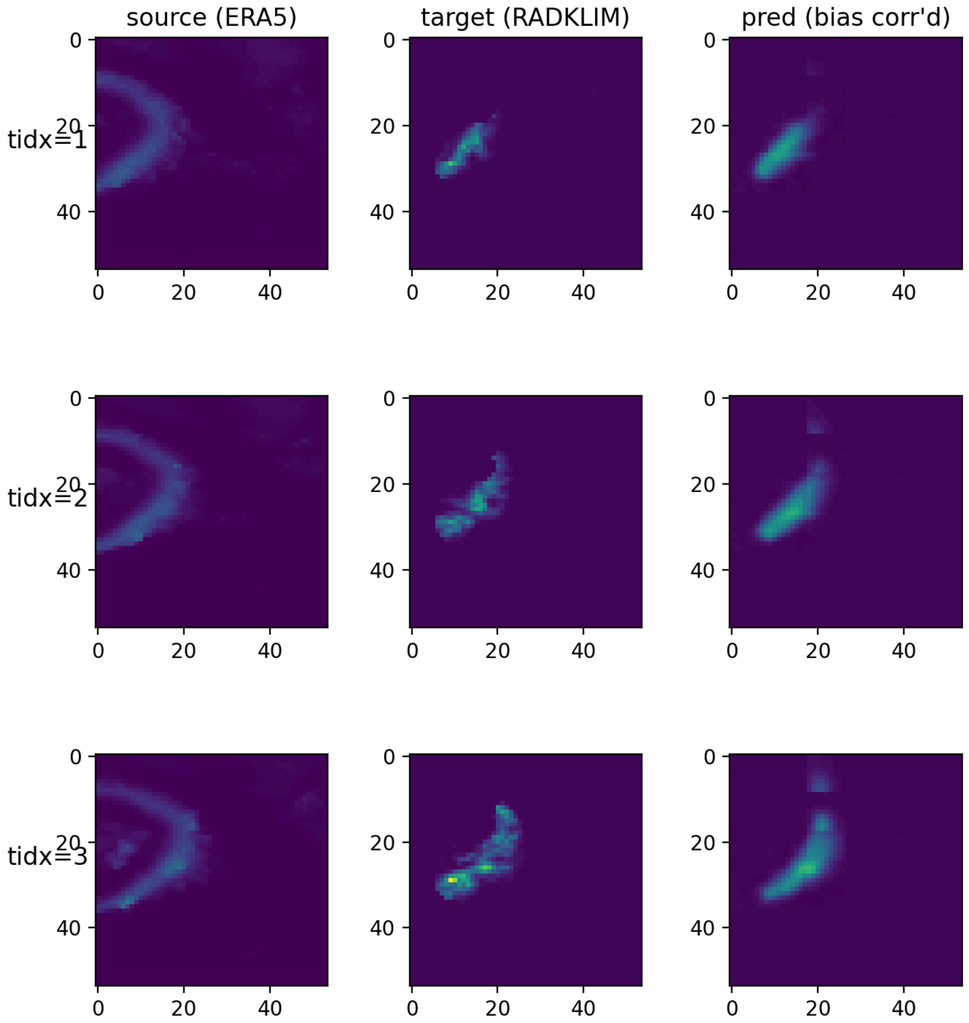
Bias correction



Bias correction



Bias correction



Outlook

Outlook

- Train an integrate many different data sources
  - CERRA, NexRAD, ...
  - IFS, ICON, ...
  - Microwave sounders, radiosondes, SYNOP, ...

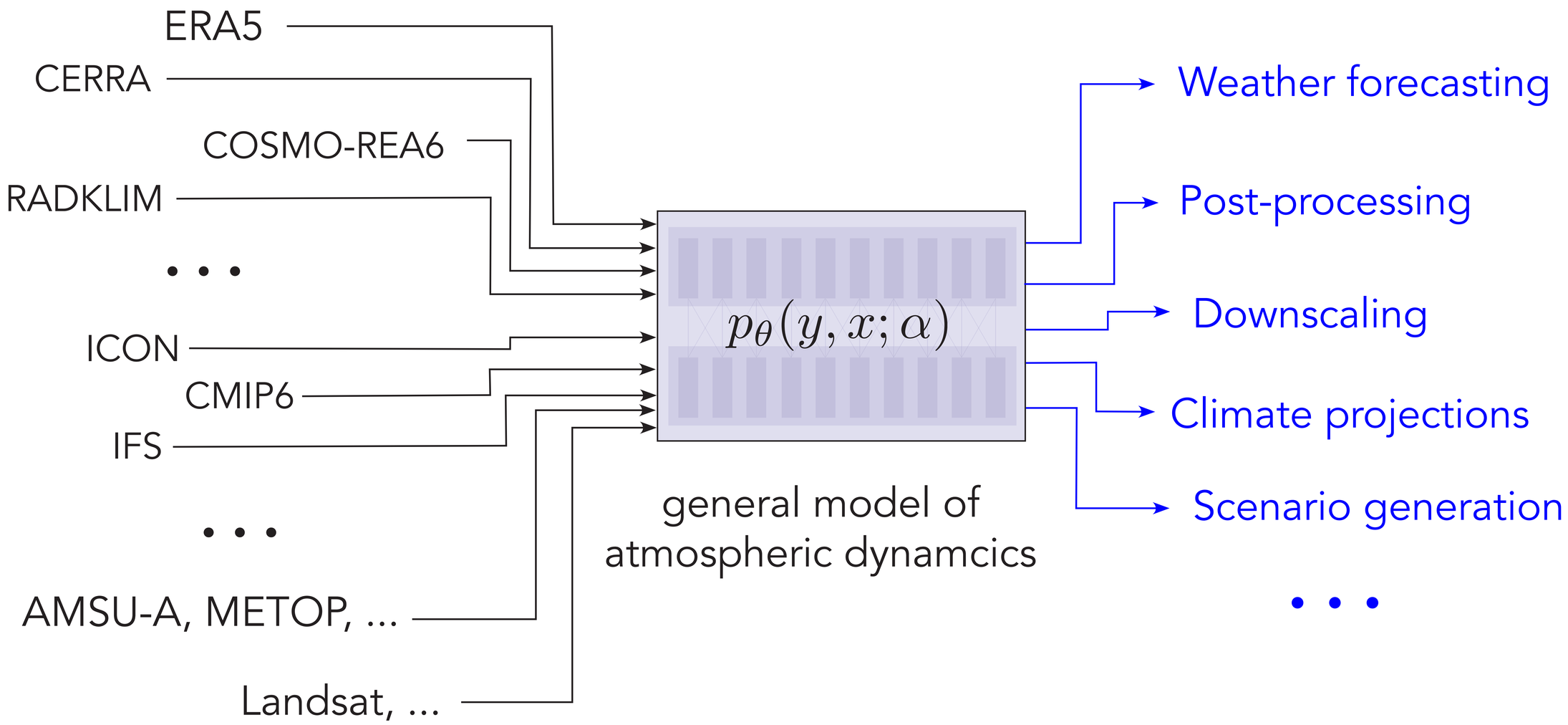
Outlook

- Train an integrate many different data sources
  - CERRA, NexRAD, ...
  - IFS, ICON, ...
  - Microwave sounders, radiosondes, SYNOP, ...
- Consider more applications
  - Medium-range weather forecasting

Outlook

- Train an integrate many different data sources
  - CERRA, NexRAD, ...
  - IFS, ICON, ...
  - Microwave sounders, radiosondes, SYNOP, ...
- Consider more applications
  - Medium-range weather forecasting
- Can one continuous update a model?

Outlook



Summary

AtmoRep

- Numerical statistical atmospheric model
  - Complementary to classical GCMs and ESMs
  - Represented by very large neural network
  - Very long training leads to continuous improvement

https://arxiv.org/abs/2308.13280

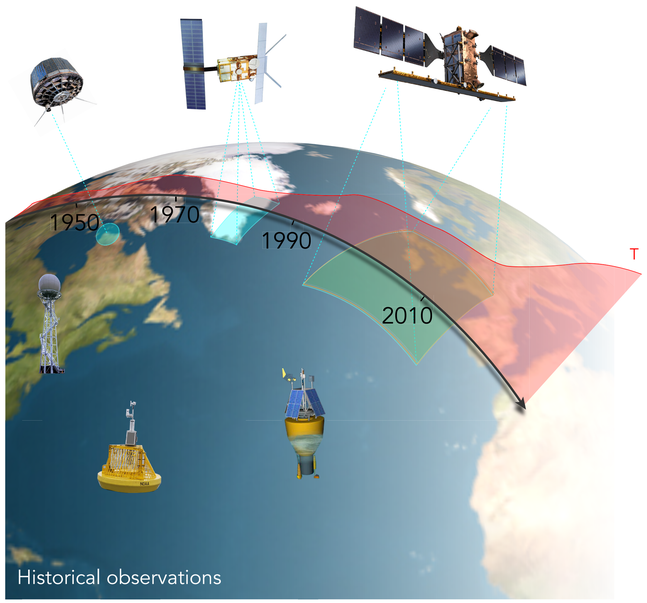
AtmoRep

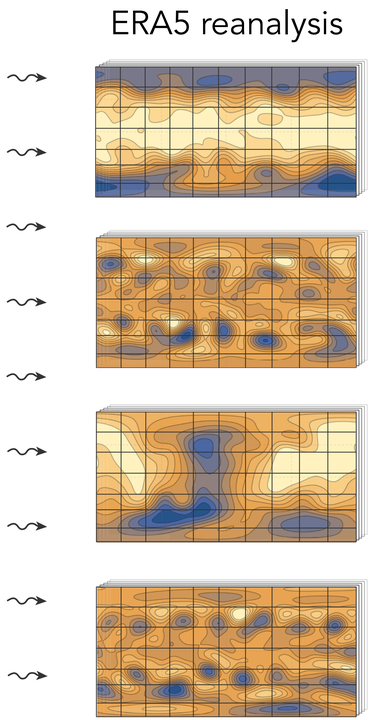
- Numerical statistical atmospheric model
  - Complementary to classical GCMs and ESMs
  - Represented by very large neural network
  - Very long training leads to continuous improvement
- BERT-type training leads to versatile intrinsic capabilites
  - Forecasting, temporal interpolation, model correction, ...
  - Straight forward extension to various applications

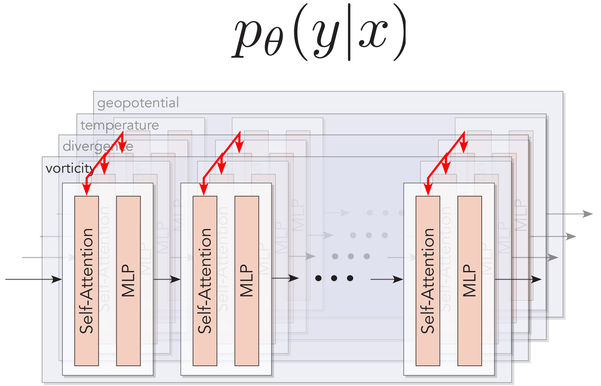
https://arxiv.org/abs/2308.13280

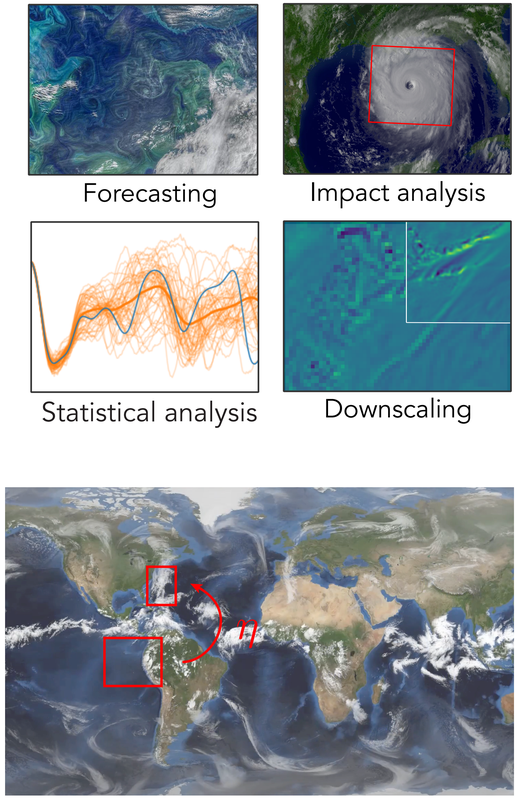
AtmoRep

https://arxiv.org/abs/2308.13280

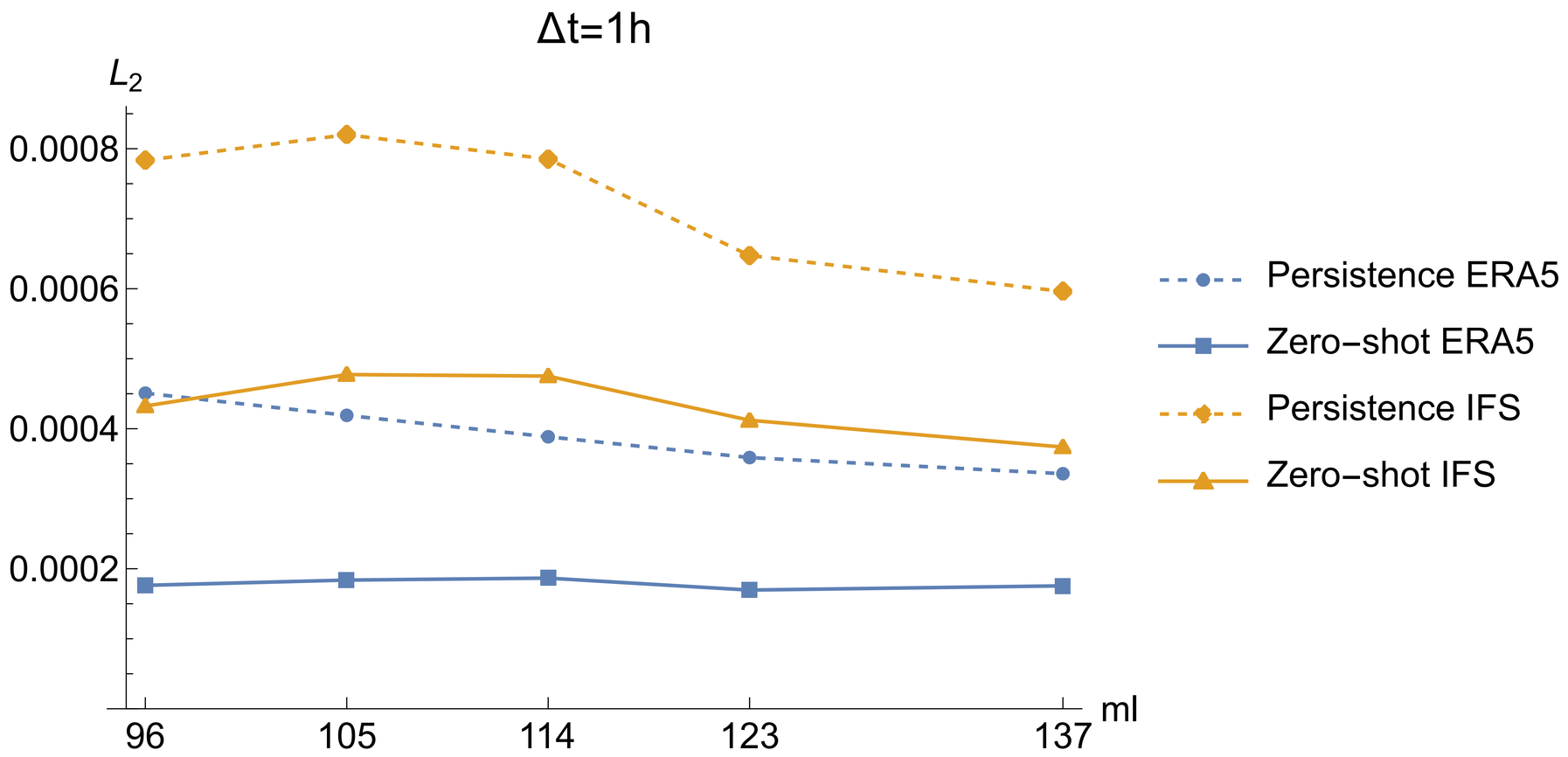




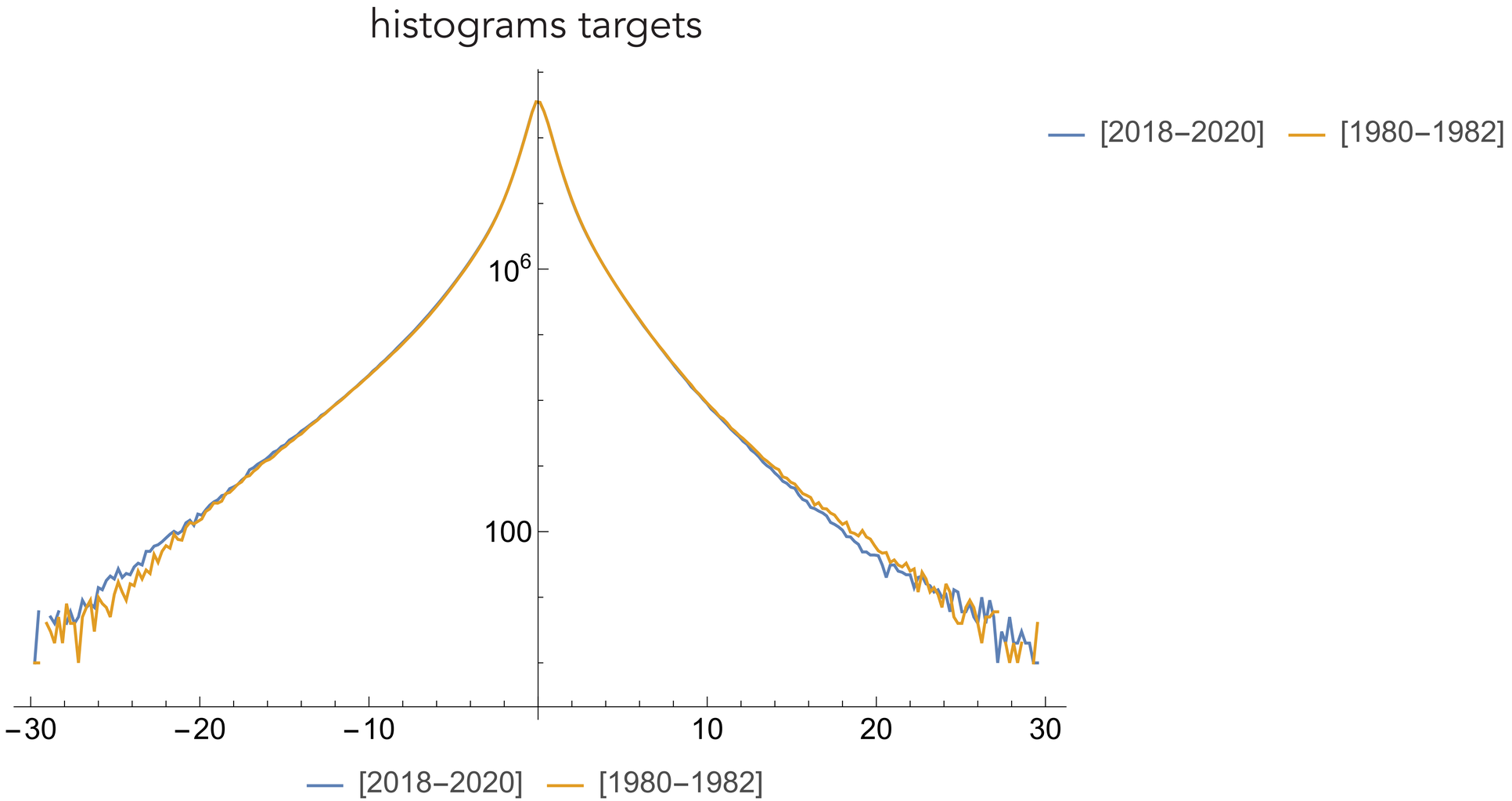




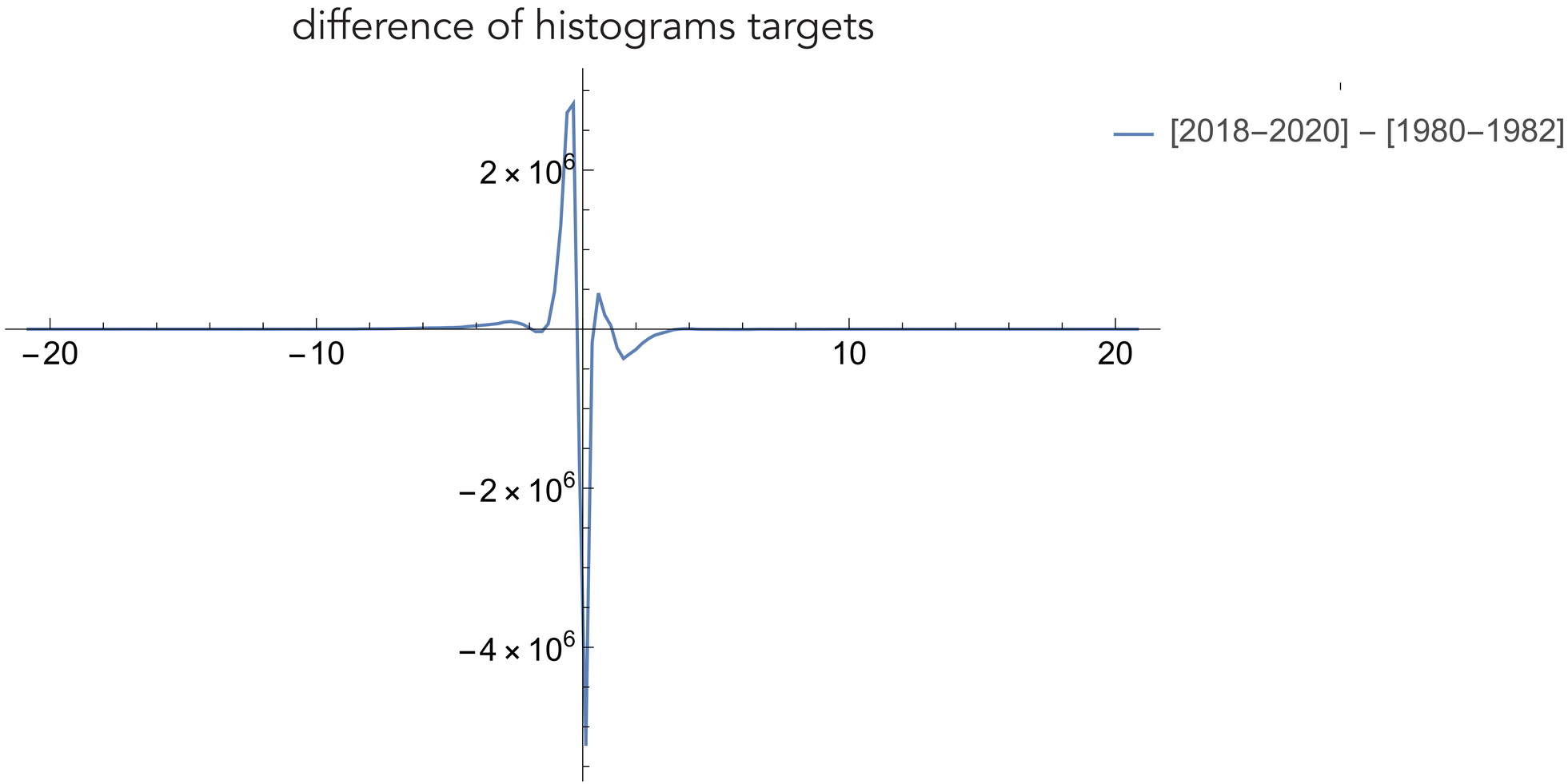
Model correction



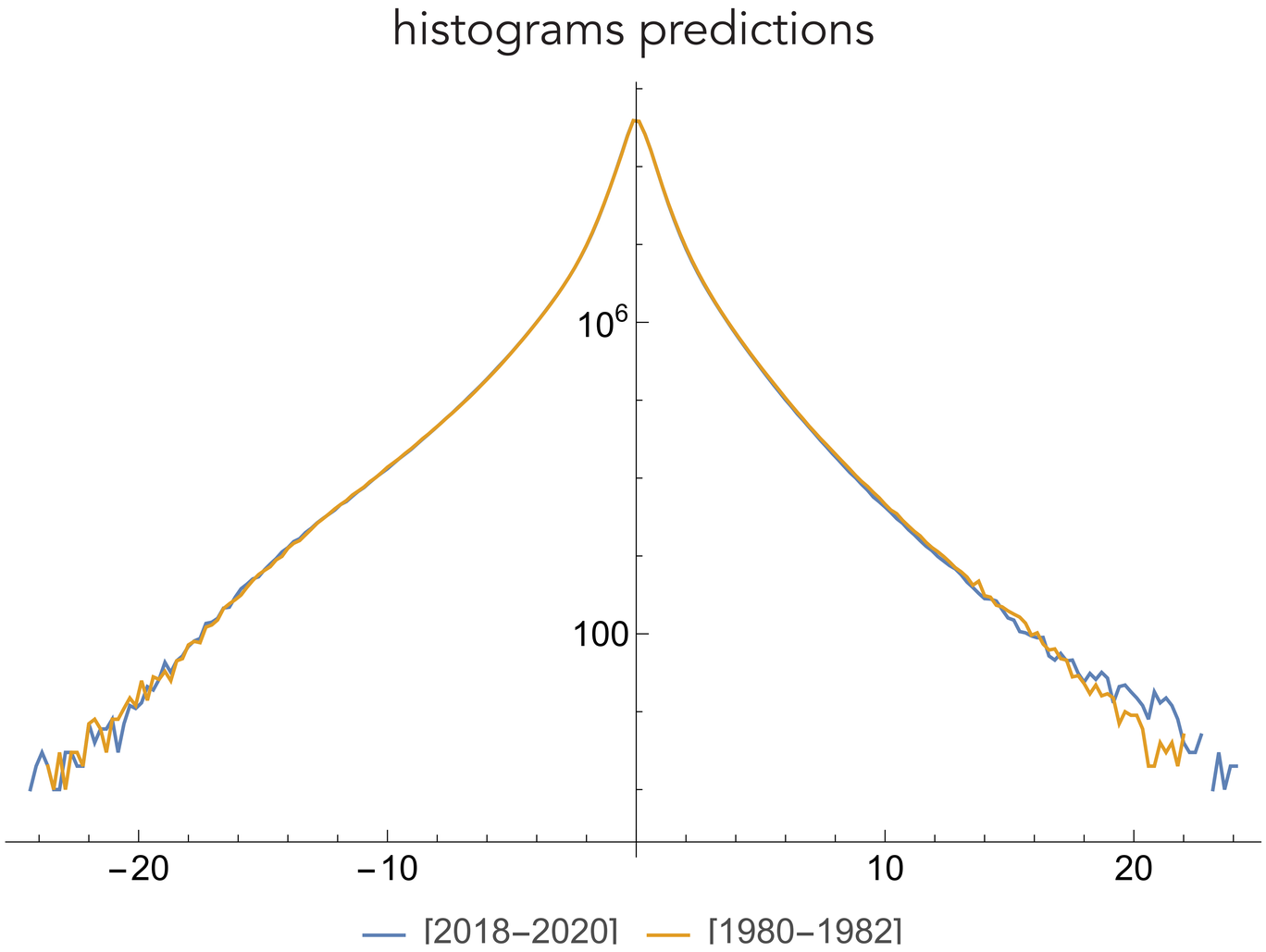
Counterfactuals



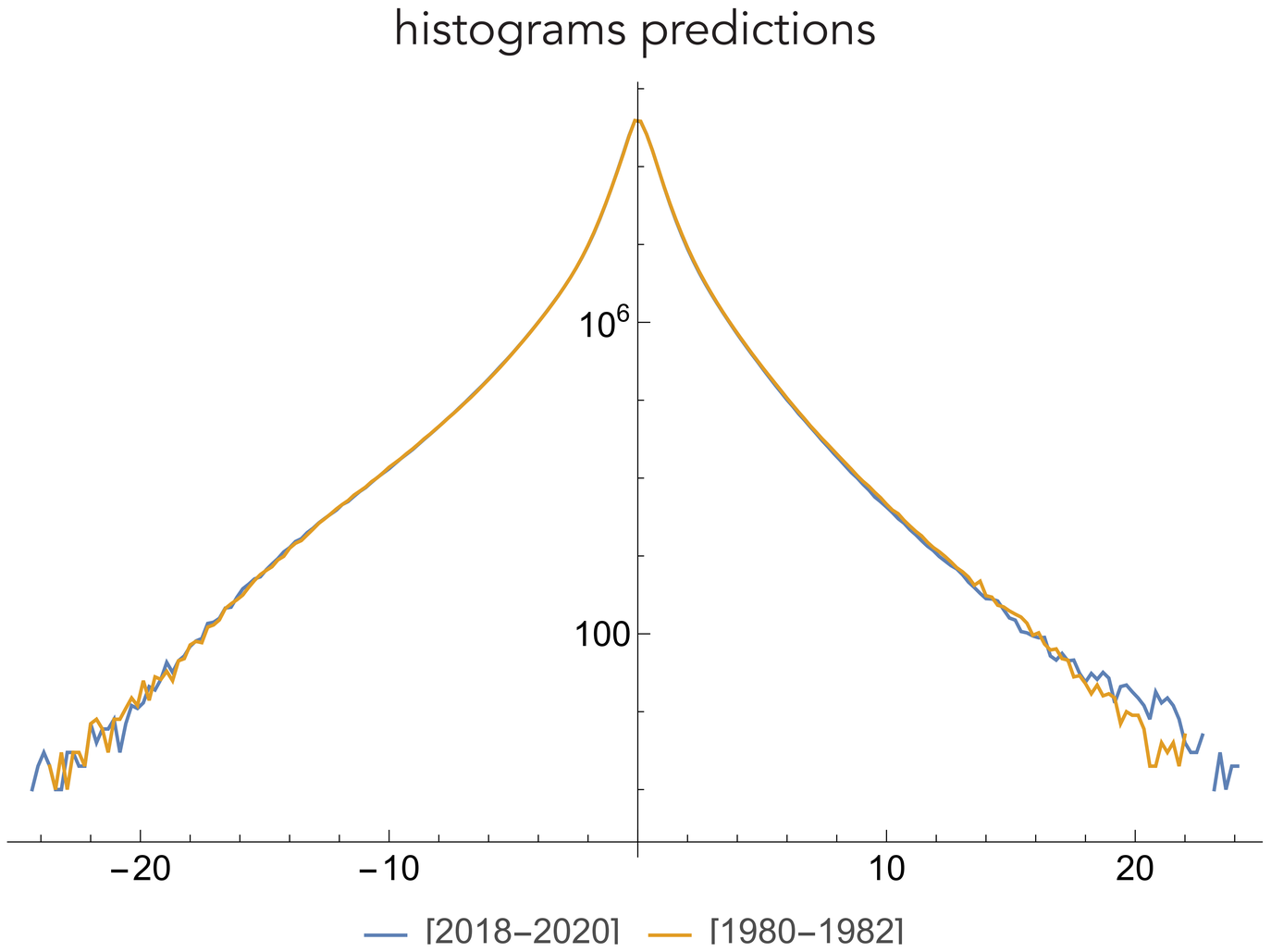
Counterfactuals



Counterfactuals



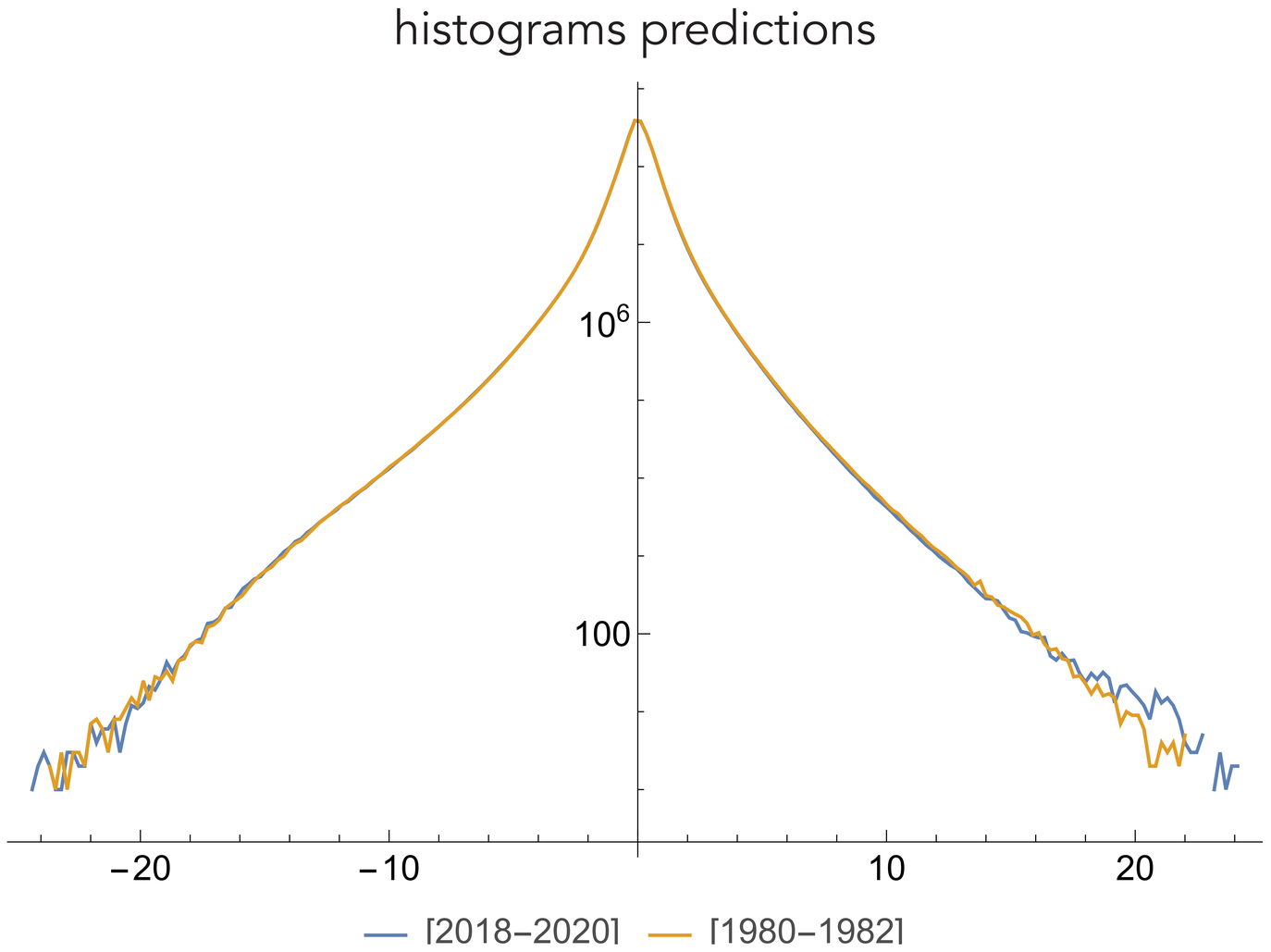
Counterfactuals



[2018–2020]

initial conditions *but*
$\alpha_{\text{year}} = [1980, 1982]$

Counterfactuals



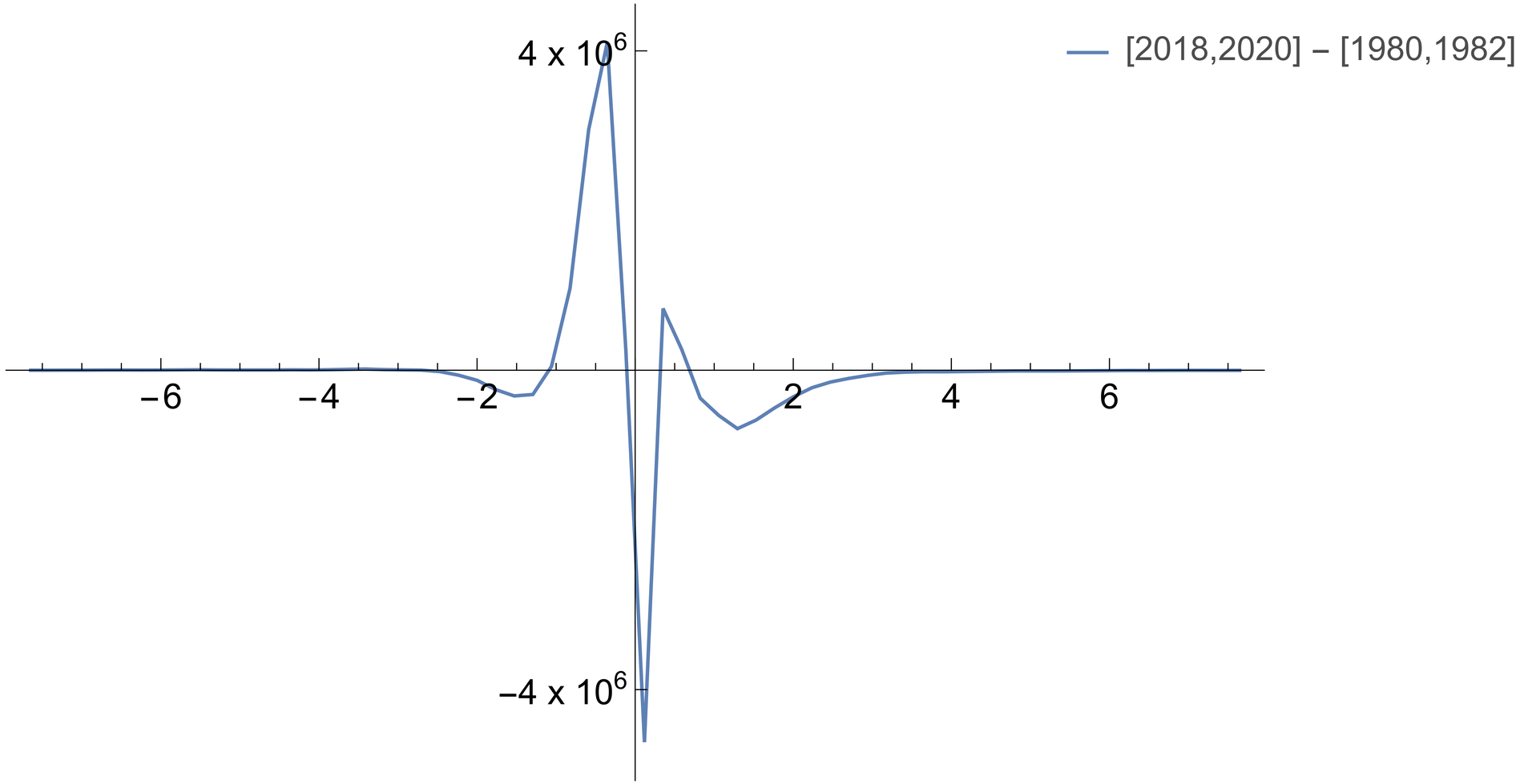
[2018–2020]

initial conditions *but* $\alpha_{\text{year}} = [1980, 1982]$

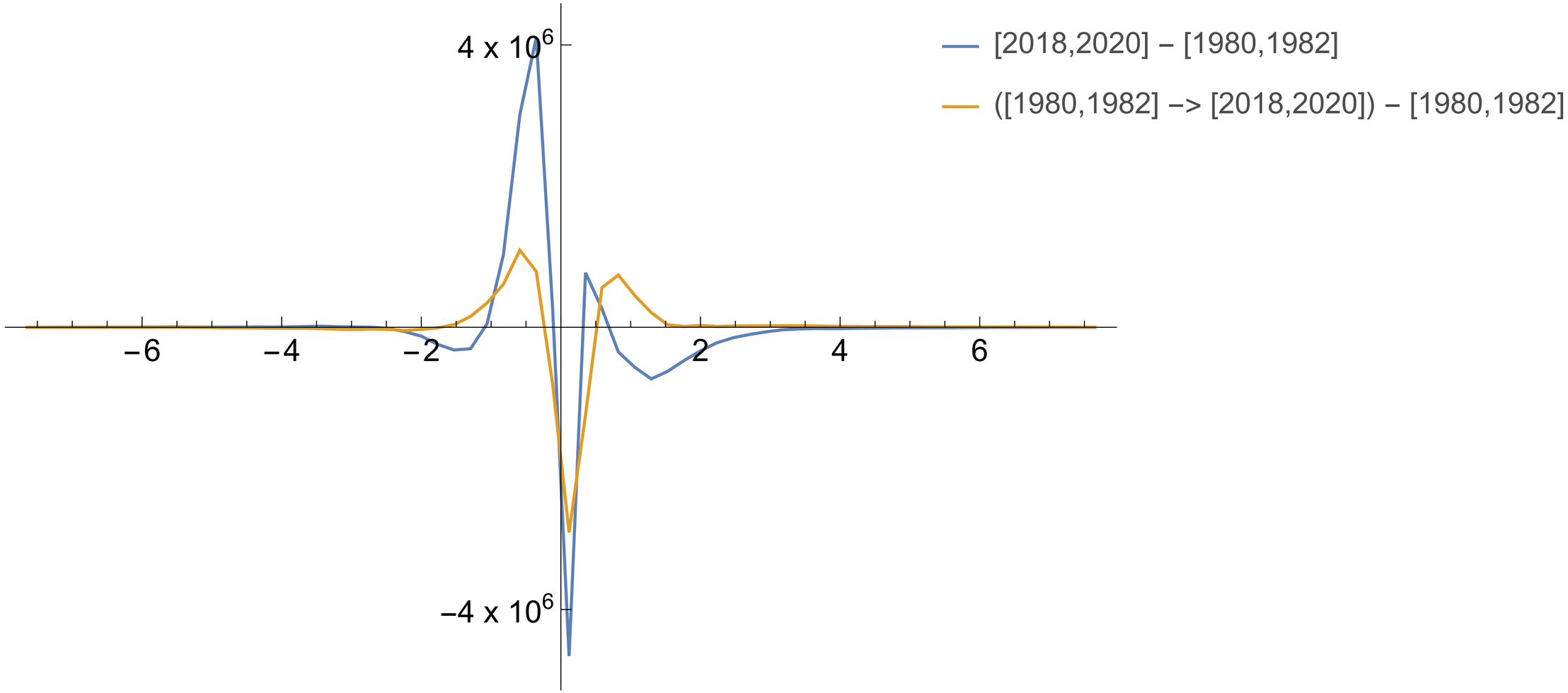
[1980–1982]

initial conditions *but* $\alpha_{\text{year}} = [2018, 2020]$

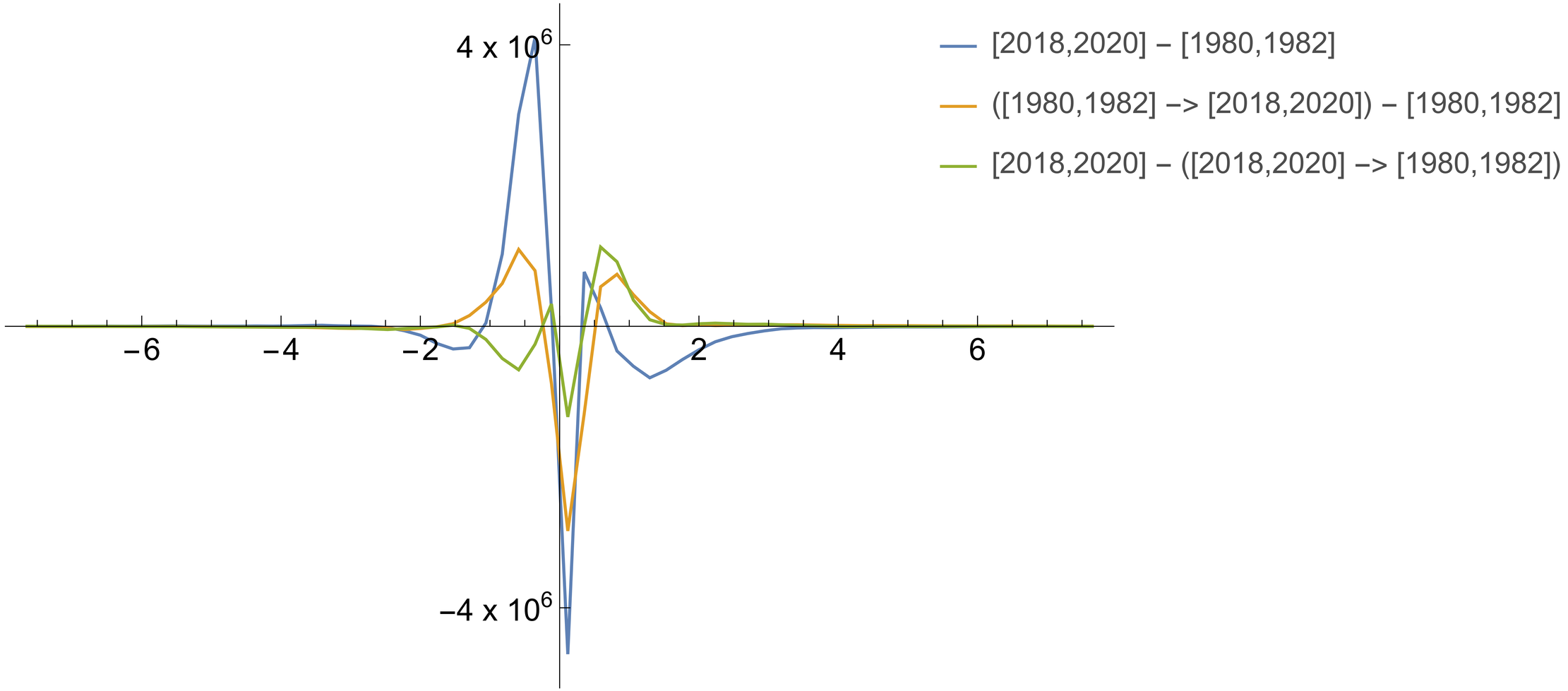
Counterfactuals



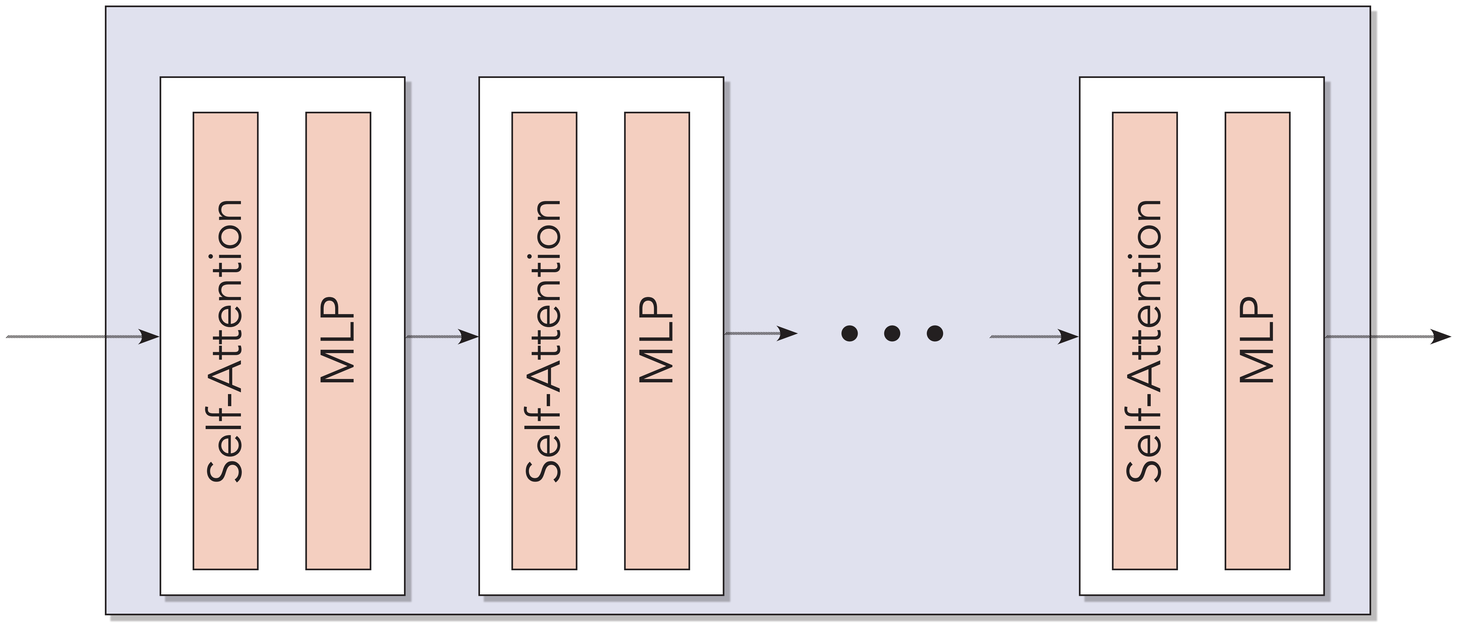
Counterfactuals



Counterfactuals



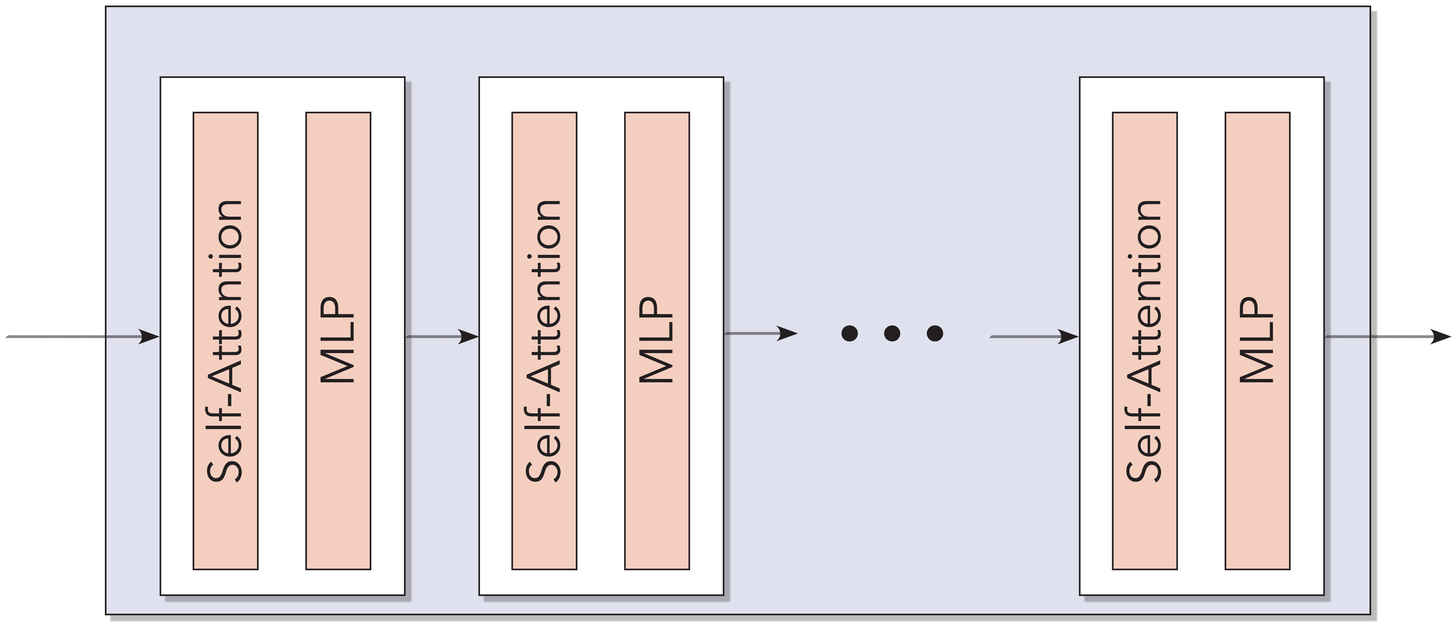
Multiformer



Multiformer

Self
attention

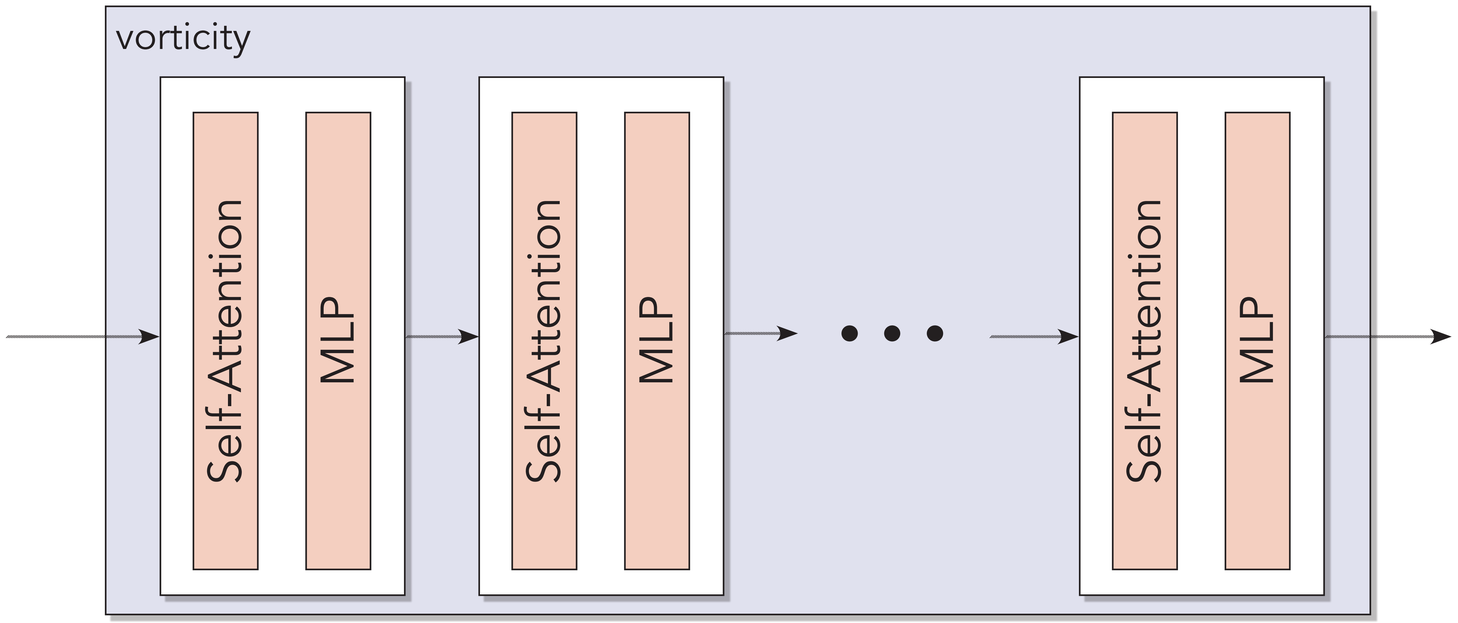
$$\sigma\left(Q\,K^T\right)V$$



Multiformer

Self
attention

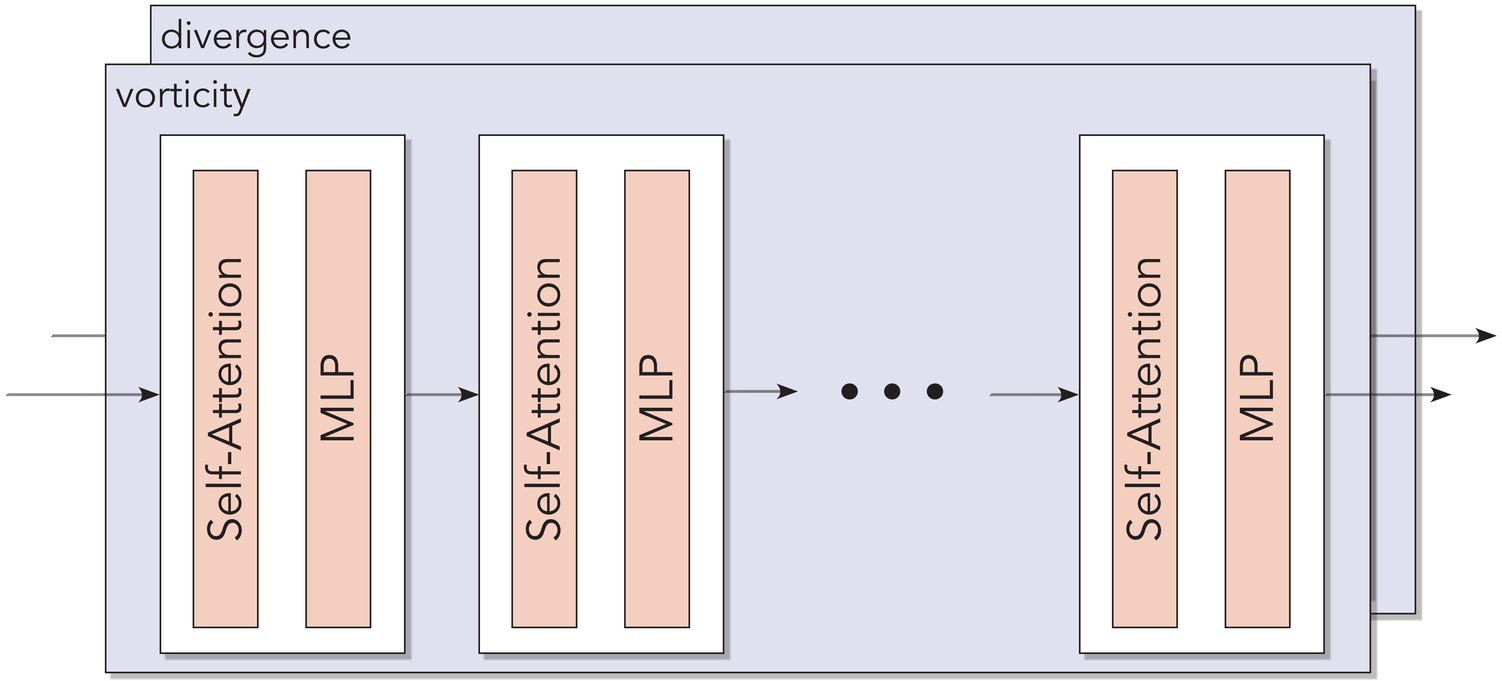
$$\sigma\left(Q\,K^T\right)V$$



Multiformer

Self
attention

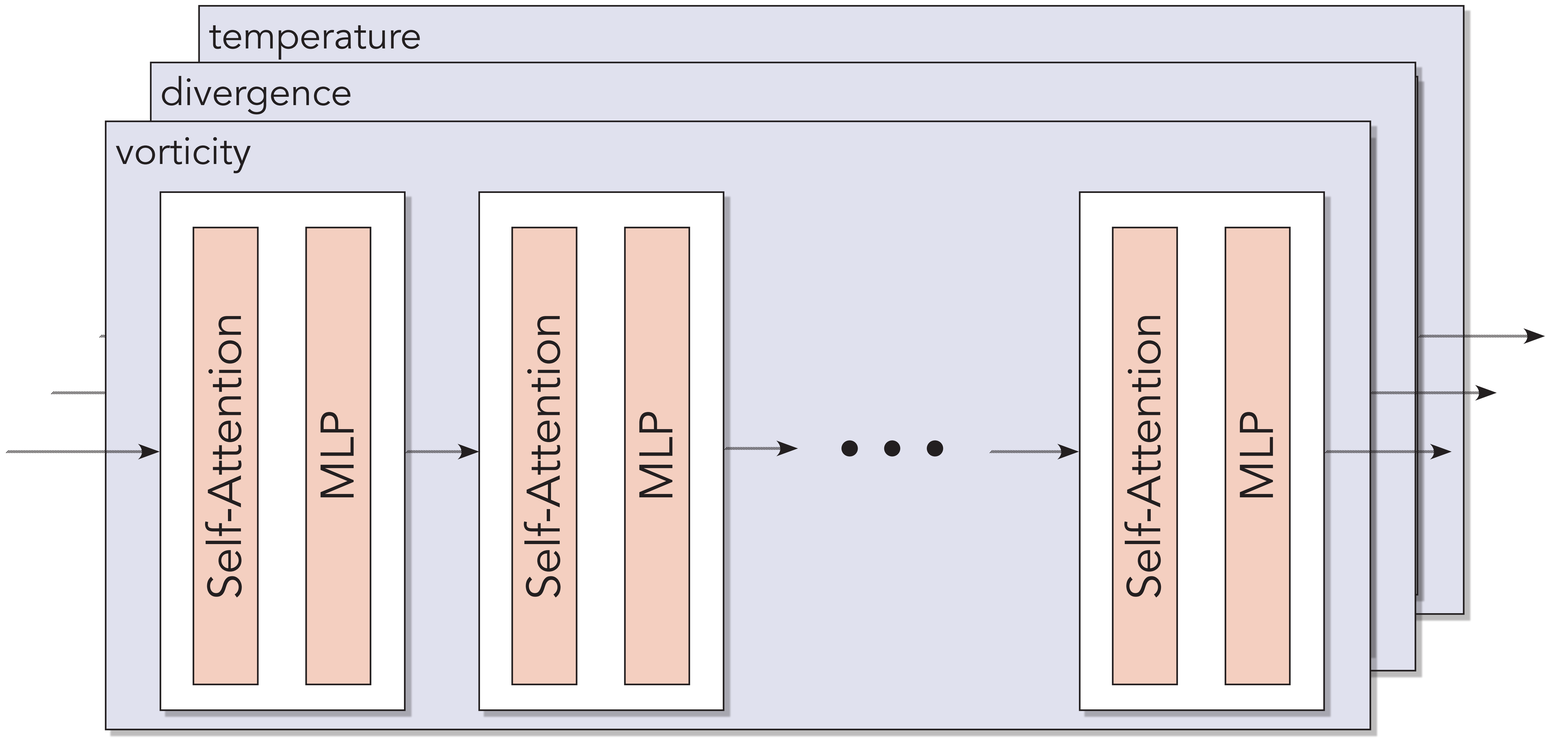
$$\sigma\left(Q\,K^T\right)V$$



Multiformer

Self
attention

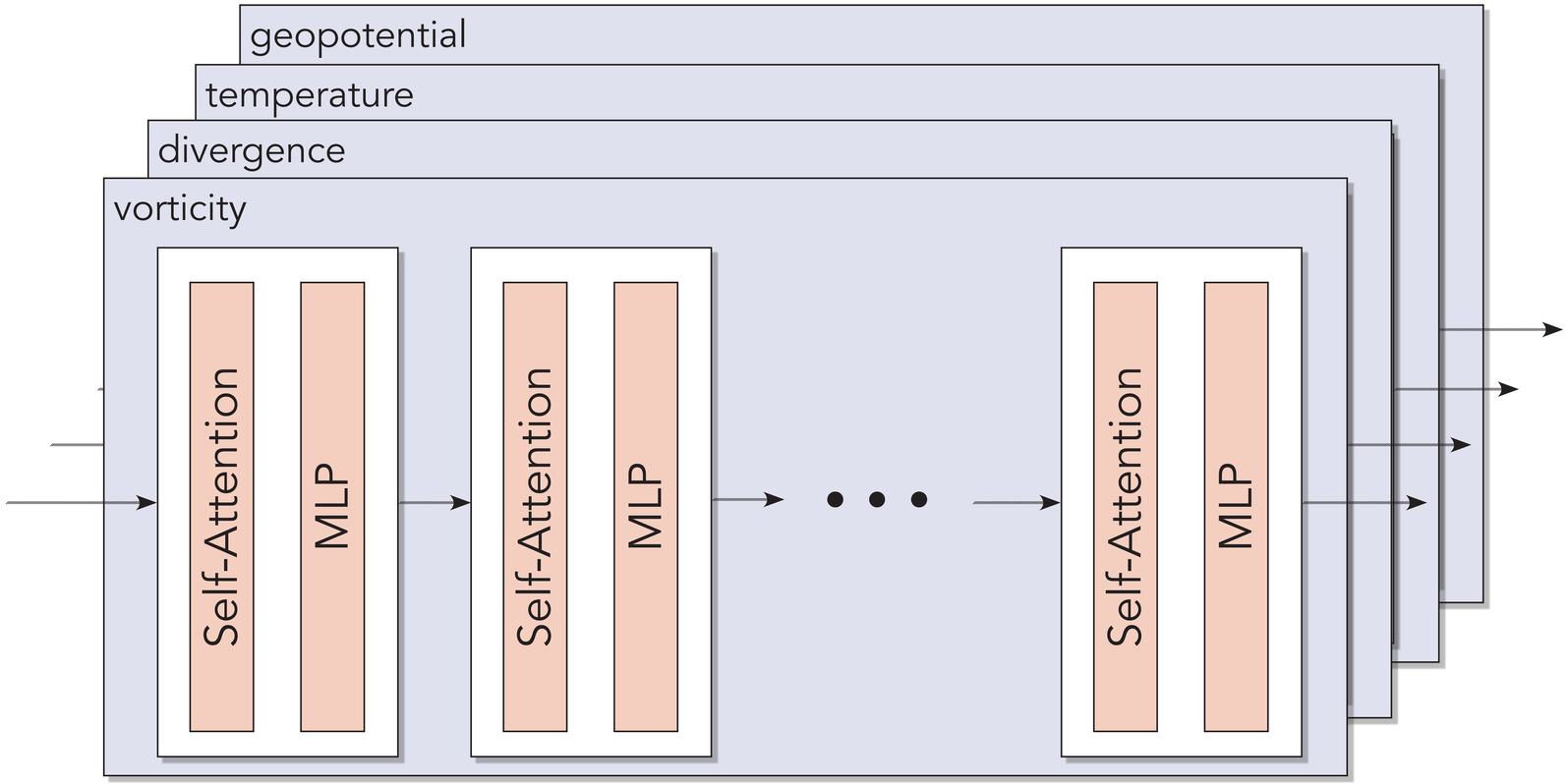
$$\sigma\left(Q\,K^T\right)V$$



Multiformer

Self
attention

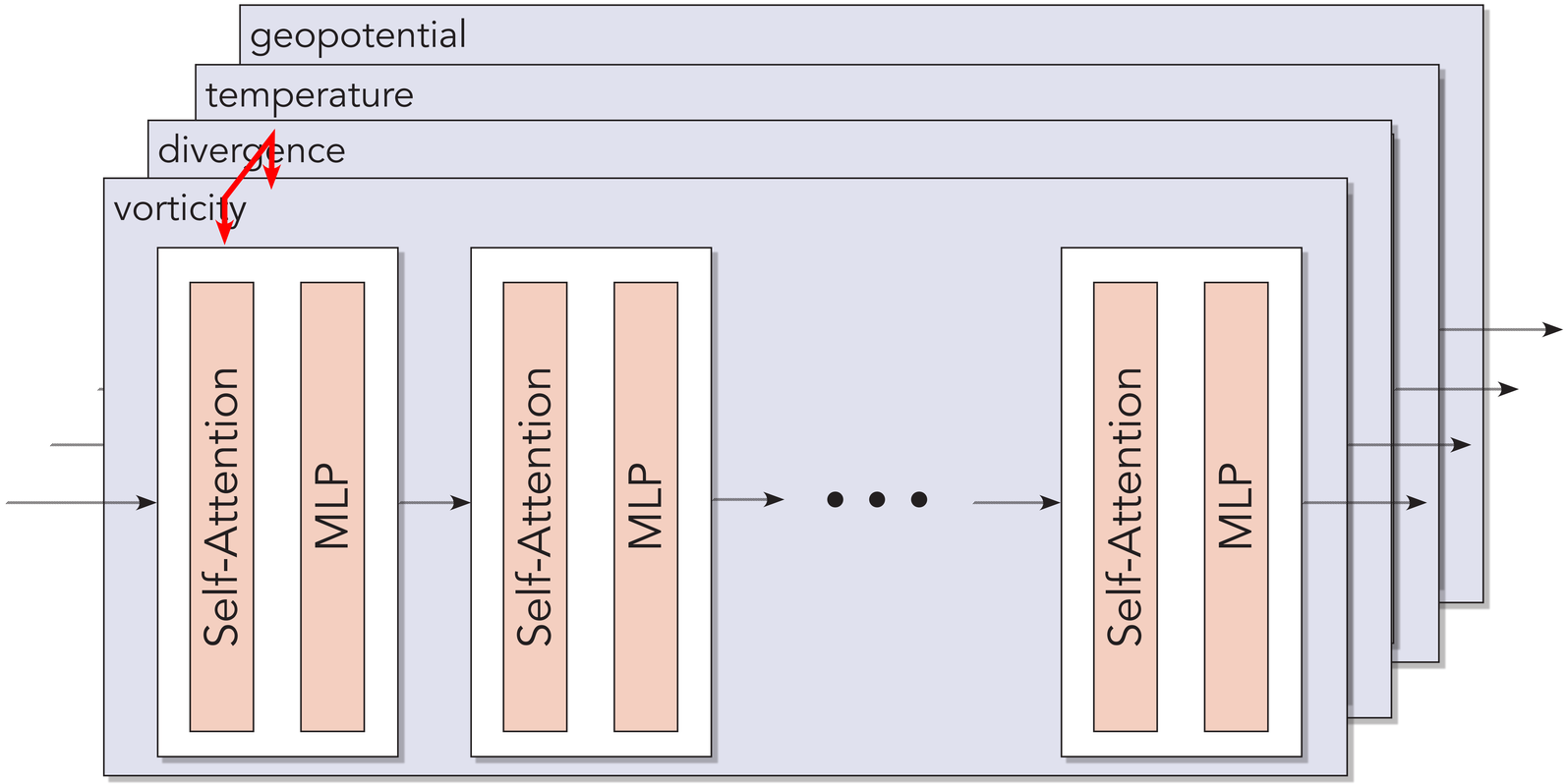
$$\sigma\left(Q\,K^T\right)V$$



Multiformer

Self
attention

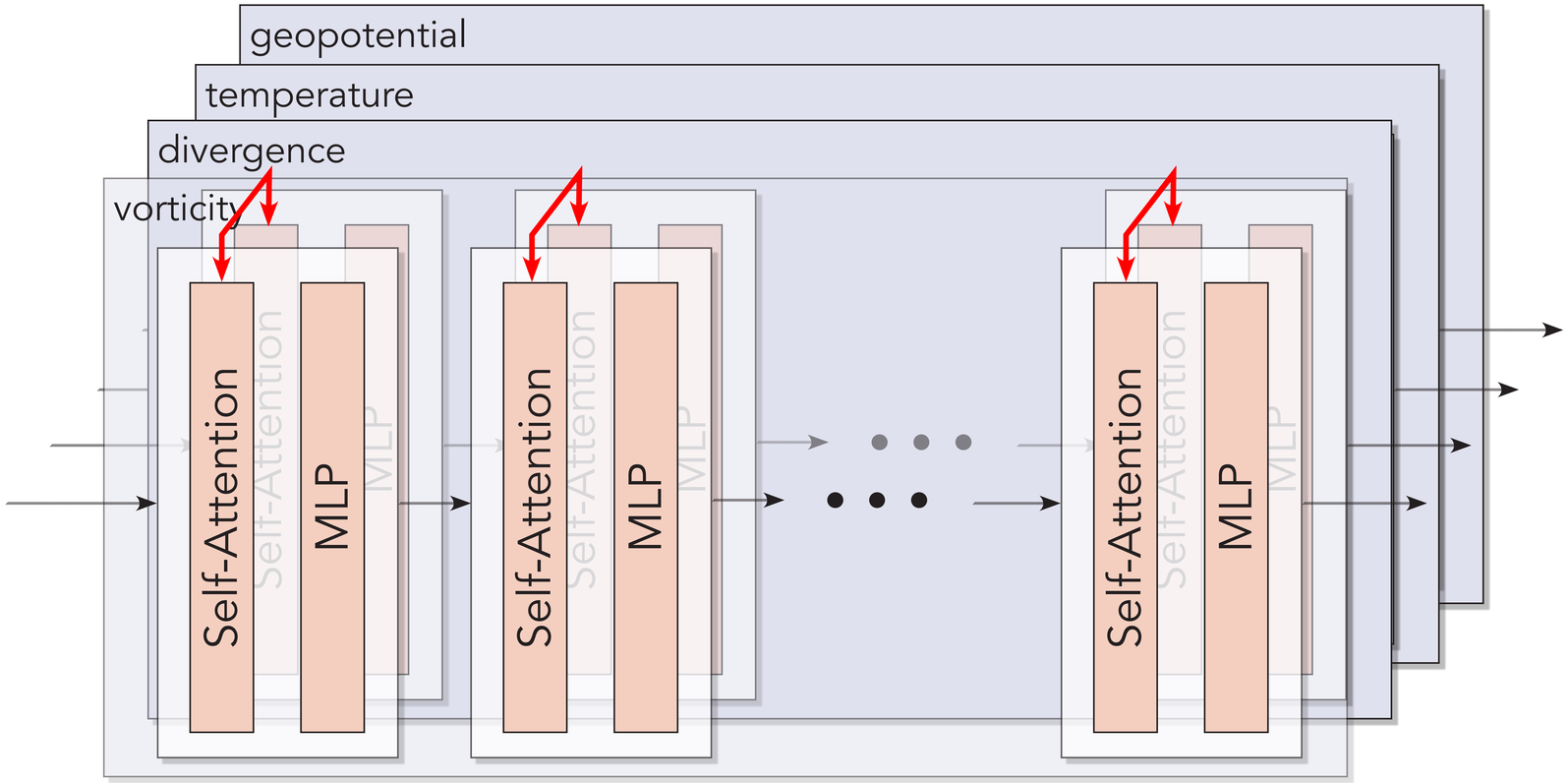
$$\sigma\left(Q\,K^T\right)V$$



Multiformer

Self
attention

$$\sigma\left(Q\,K^T\right)V$$



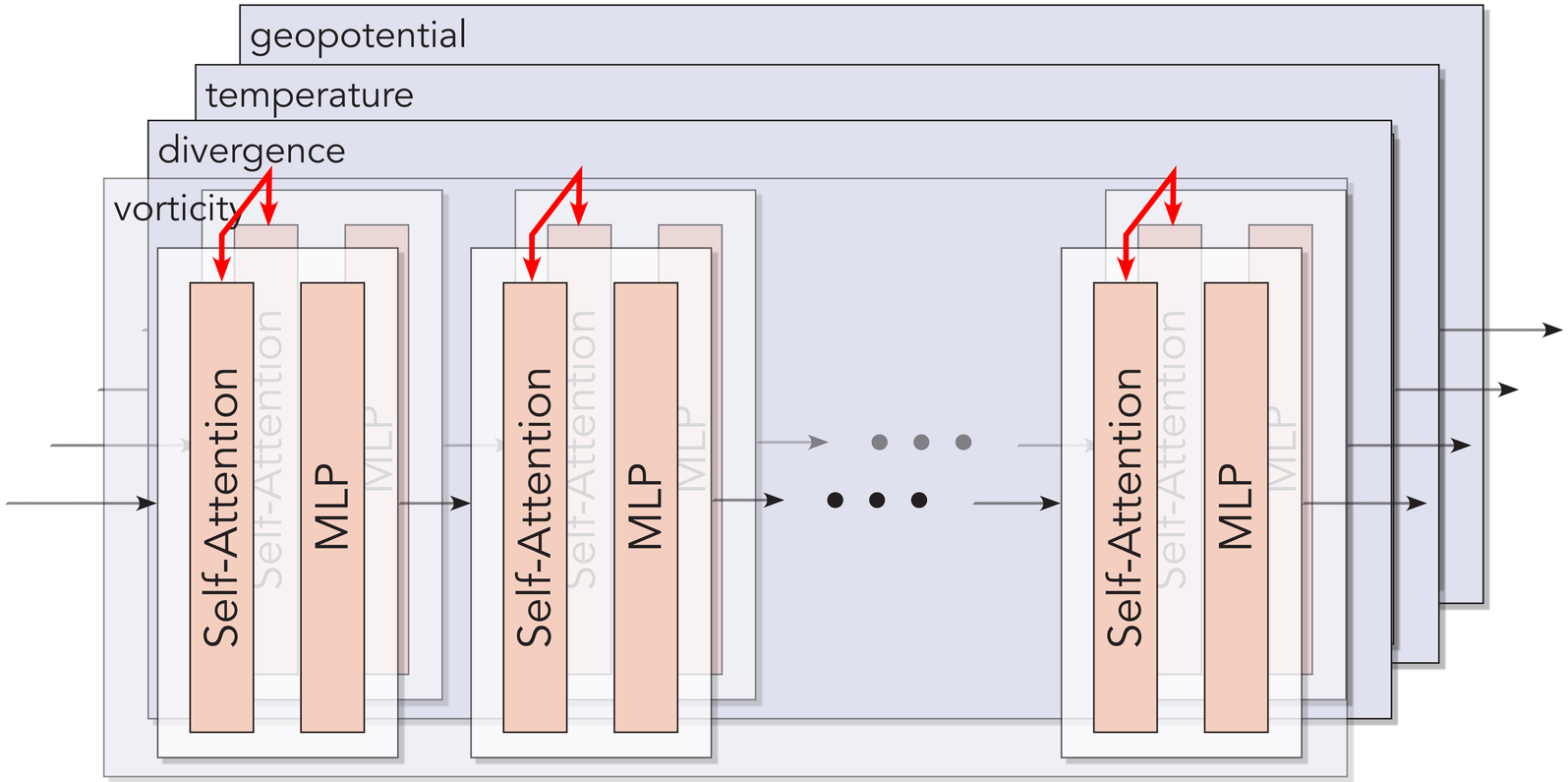
Multiformer

Self attention

$$\sigma\left(Q\,K^T\right)V$$

Cross attention

$$\sigma\left(Q_\zeta\,K_\mu^T\right)V_\mu$$



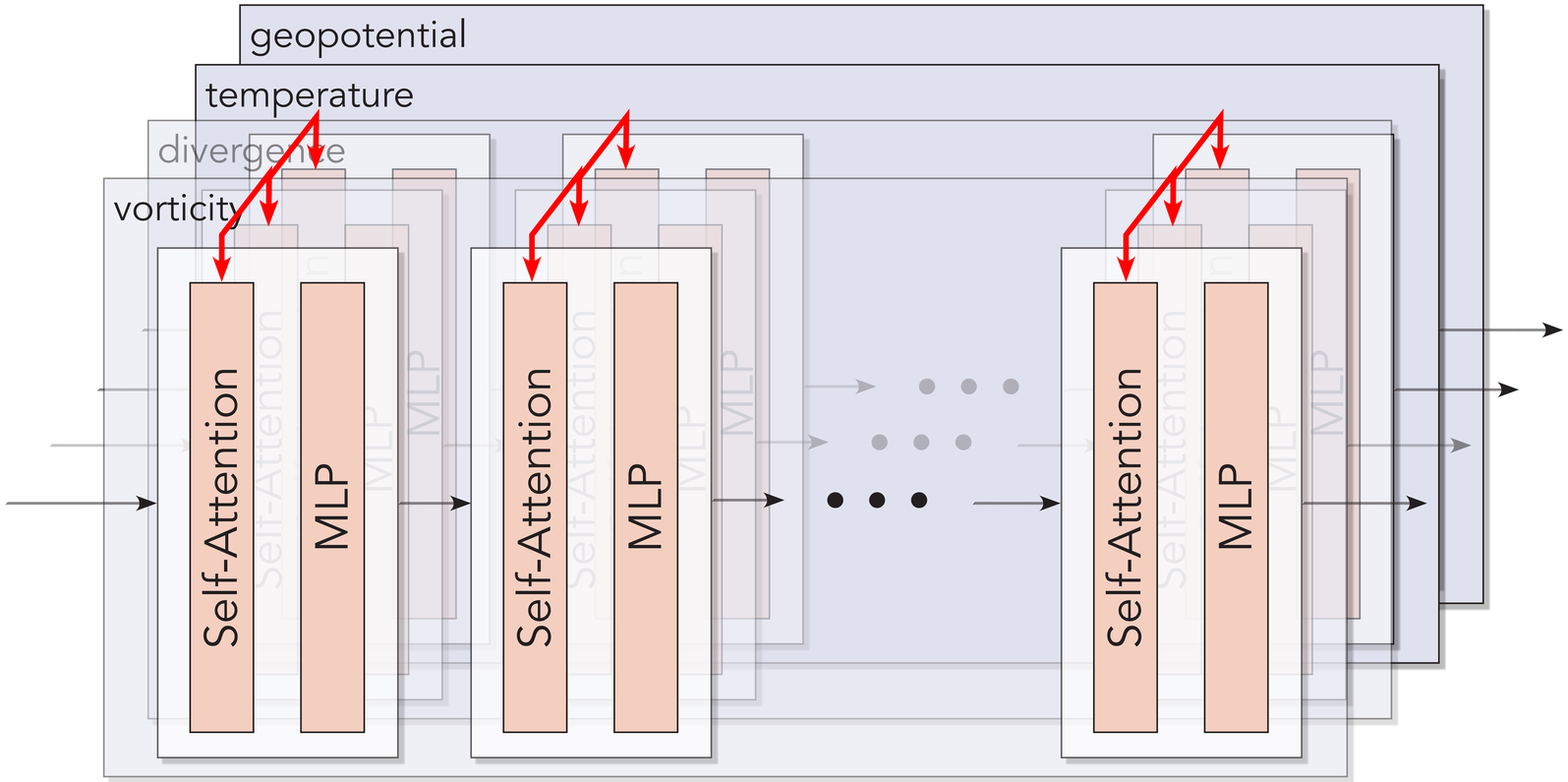
Multiformer

Self attention

$$\sigma\left(Q\,K^T\right)V$$

Cross attention

$$\sigma\left(Q_\zeta\,K_\mu^T\right)V_\mu$$



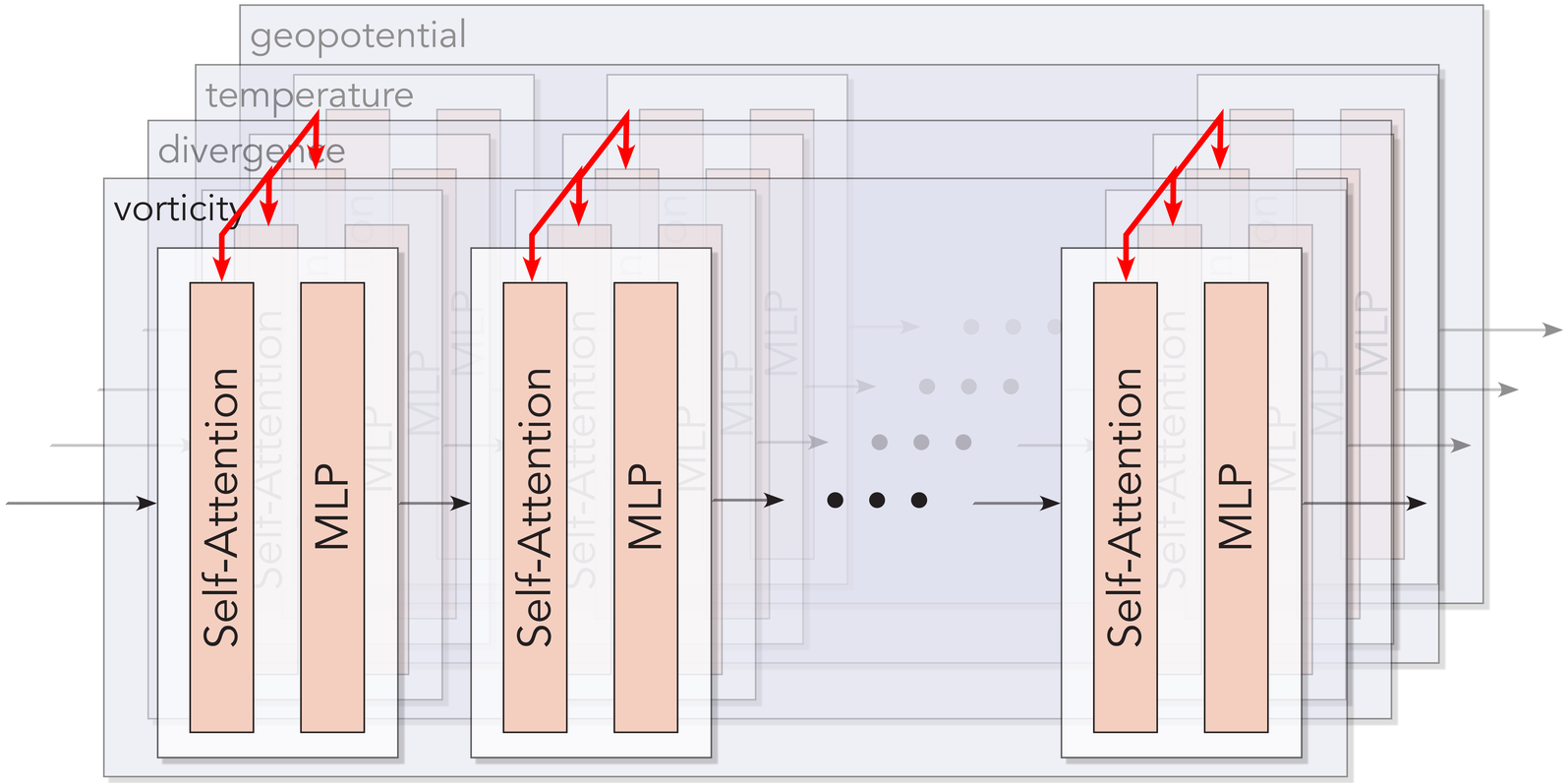
Multiformer

Self attention

$$\sigma\left(Q\,K^T\right)V$$

Cross attention

$$\sigma\left(Q_\zeta\,K_\mu^T\right)V_\mu$$



Multiformer

- Plug-and-play of fields
  - Fields can be added/removed with limited (or no) computational effort

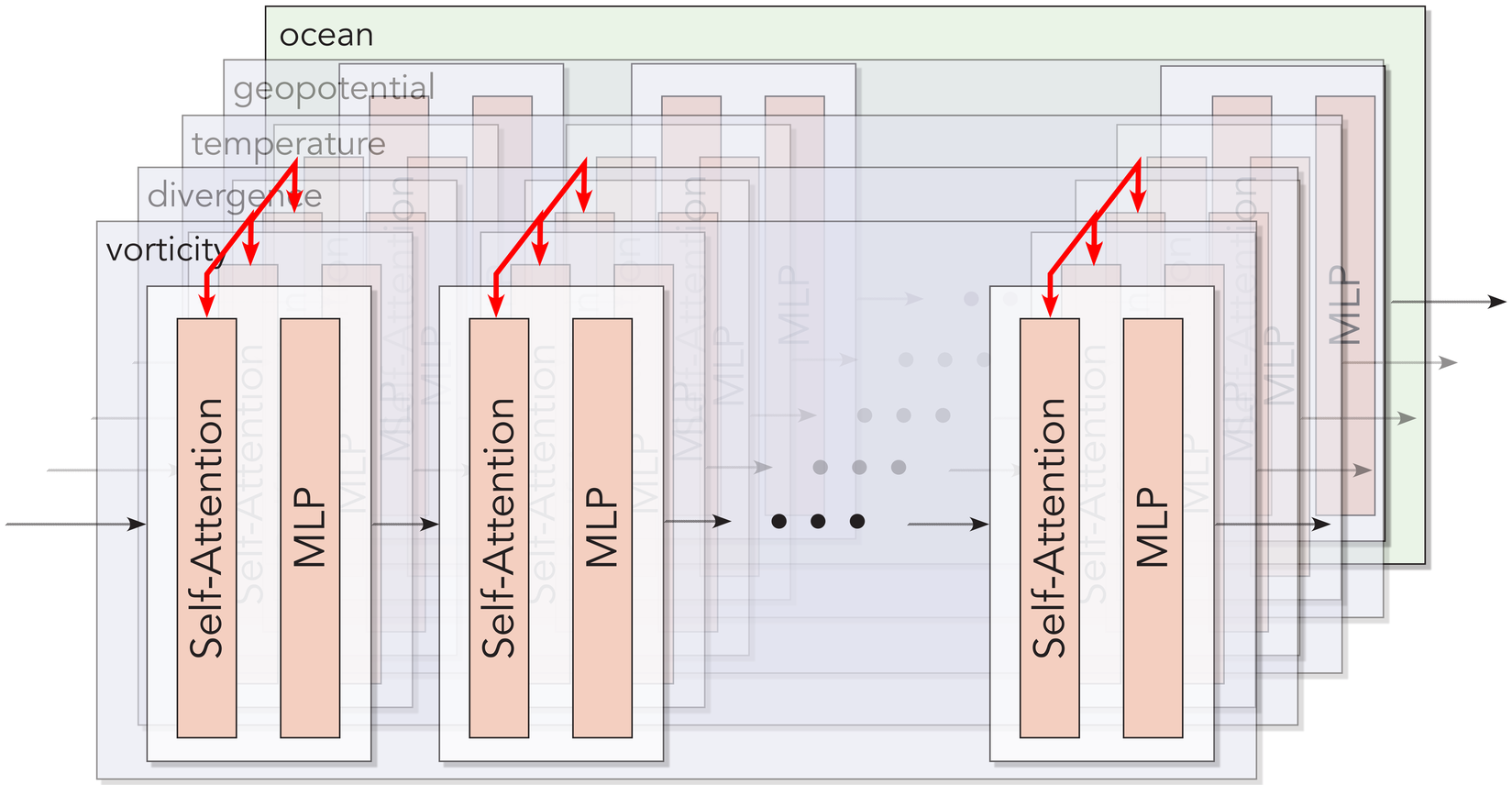
Multiformer

Self attention

$$\sigma\left(Q\,K^T\right)V$$

Cross attention

$$\sigma\left(Q_\zeta\,K_\mu^T\right)V_\mu$$



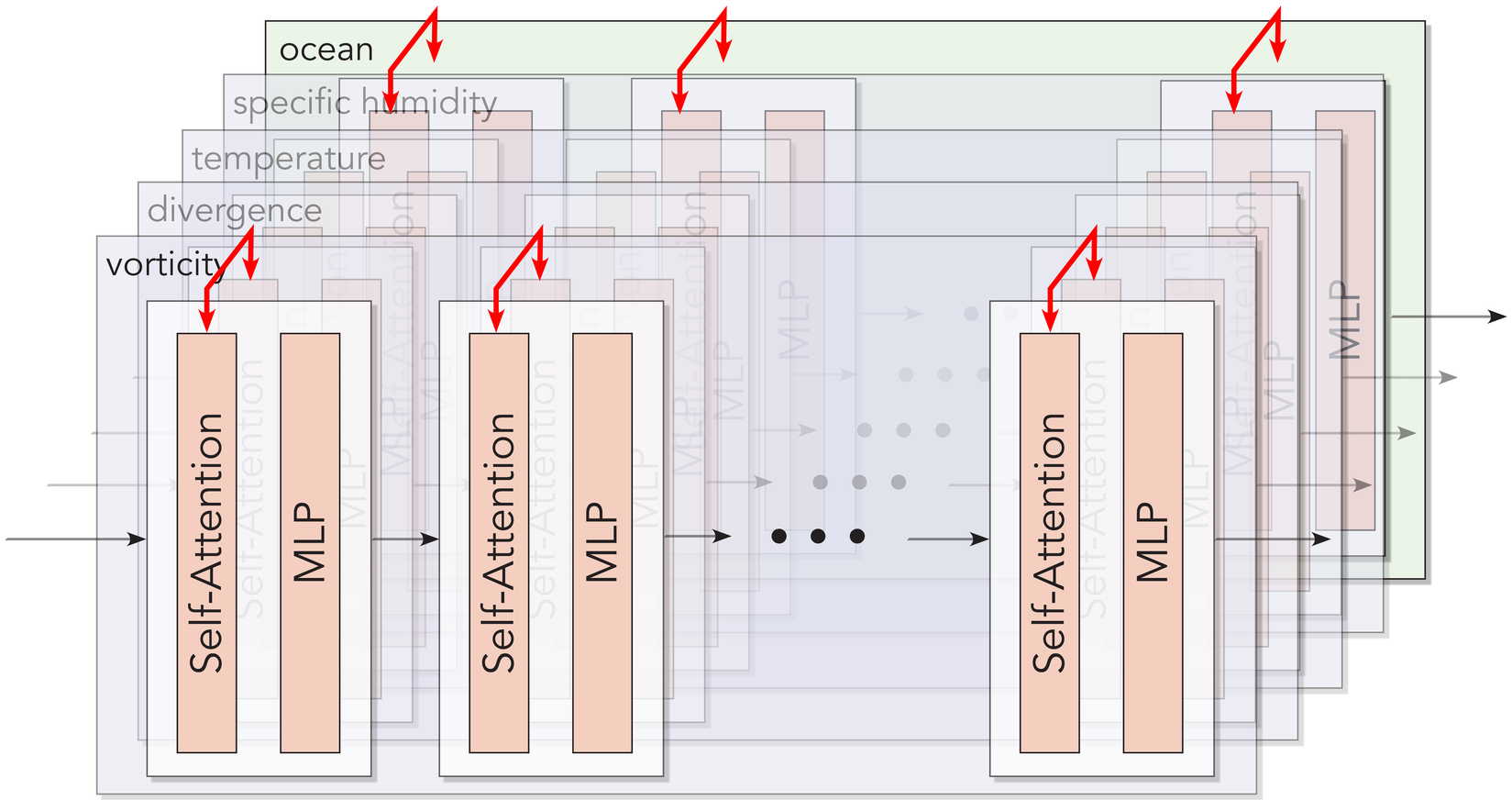
Multiformer

Self attention

$$\sigma\left(Q\,K^T\right)V$$

Cross attention

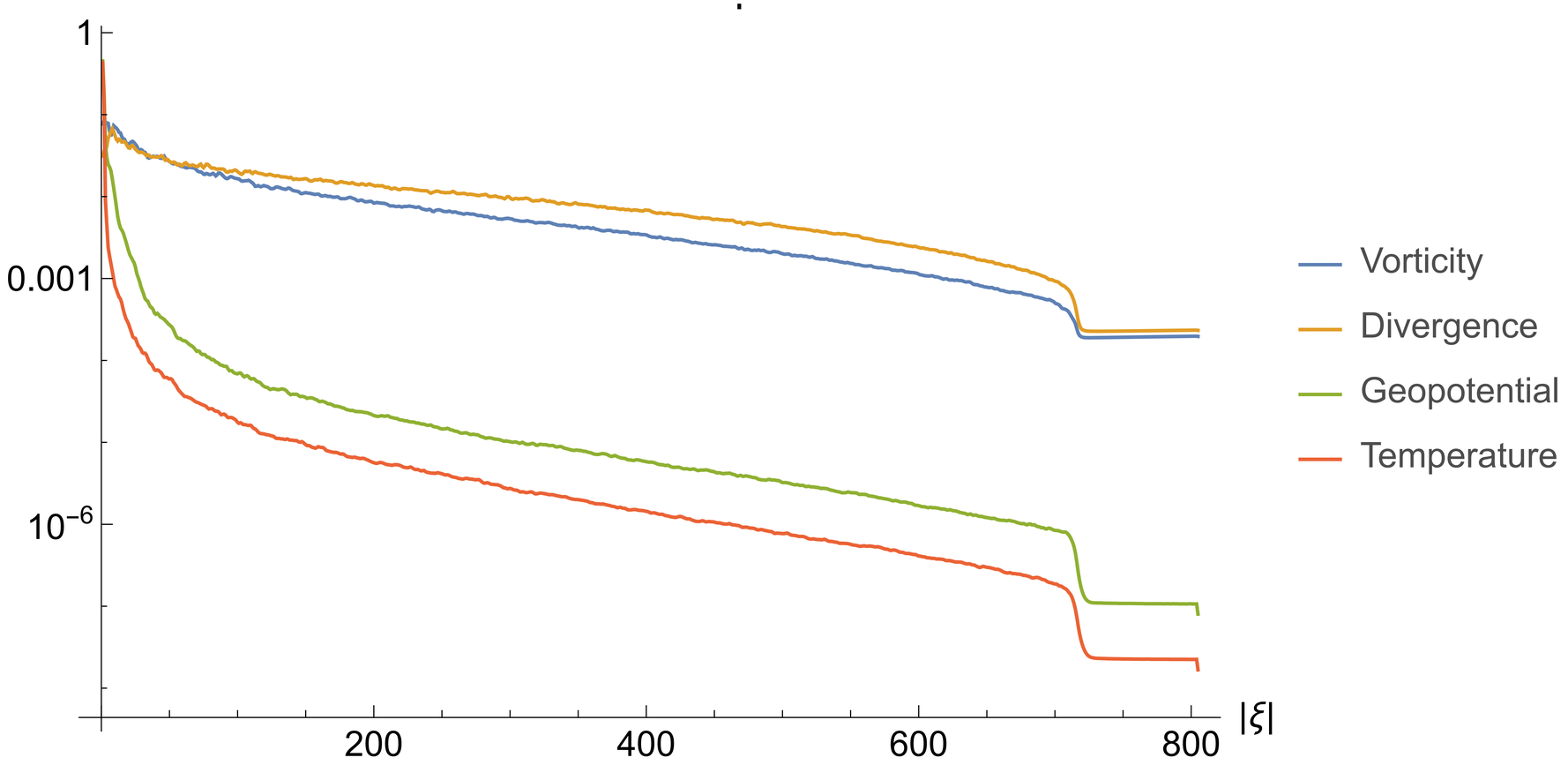
$$\sigma\left(Q_\zeta\,K_\mu^T\right)V_\mu$$



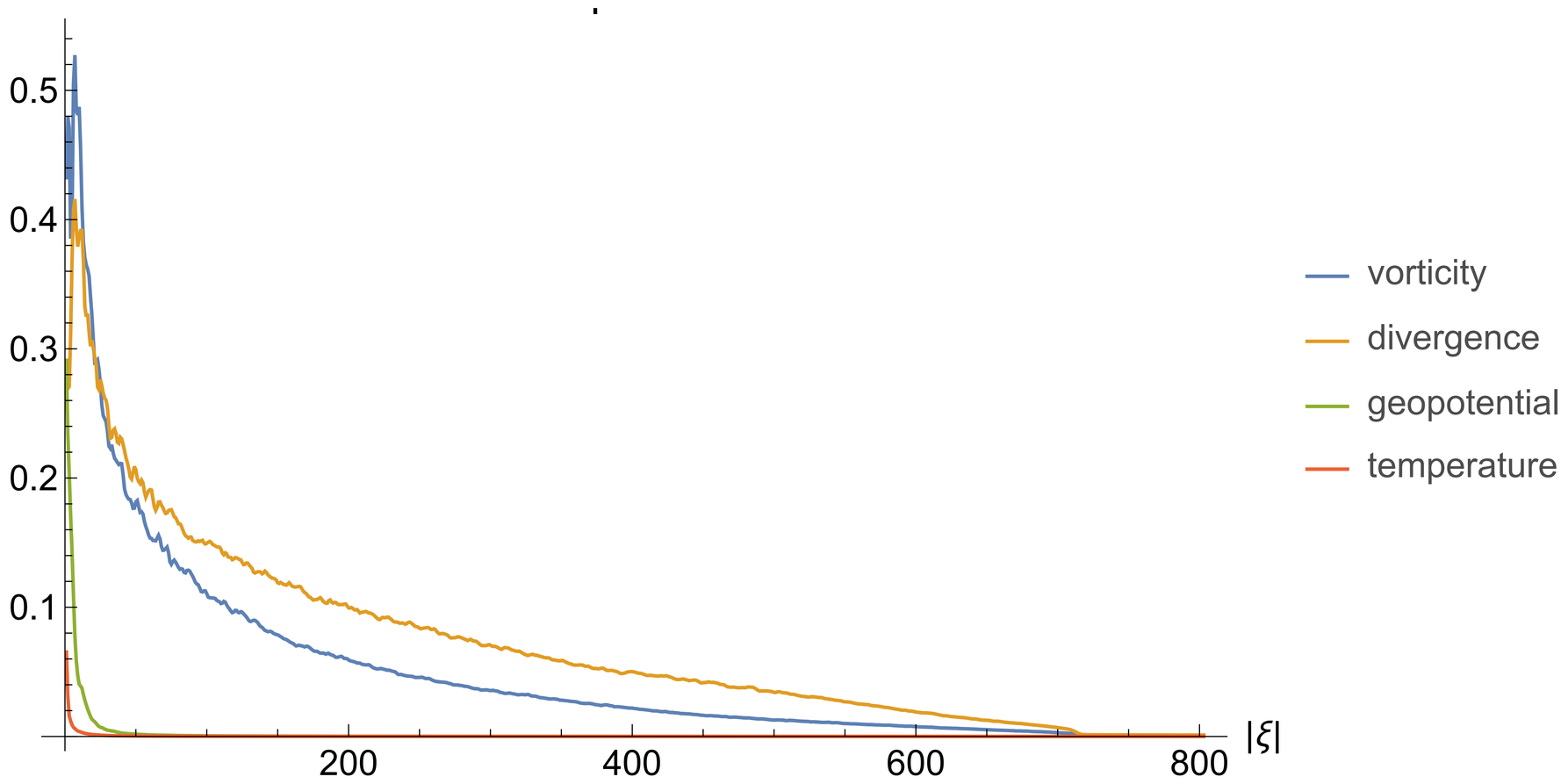
Multiformer

- Plug-and-play of fields
  - Fields can be added/removed with very limited computational effort
- Cross-attention allows for explicit introspection of interaction between fields
- Different physical fields with different properties have separate latent spaces (and transformations for these)

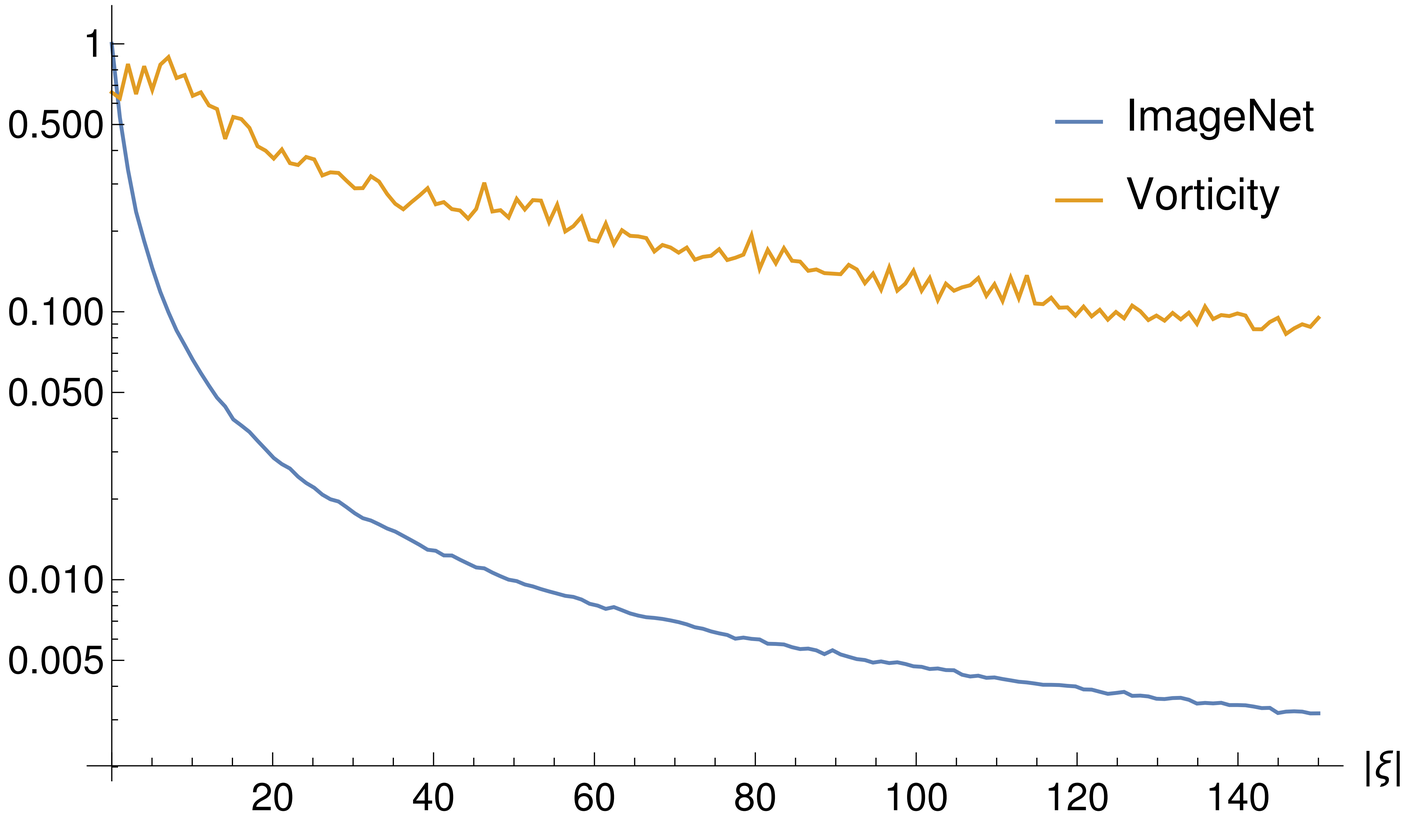
AtmoRep data



AtmoRep data



ERA5 versus ImageNet



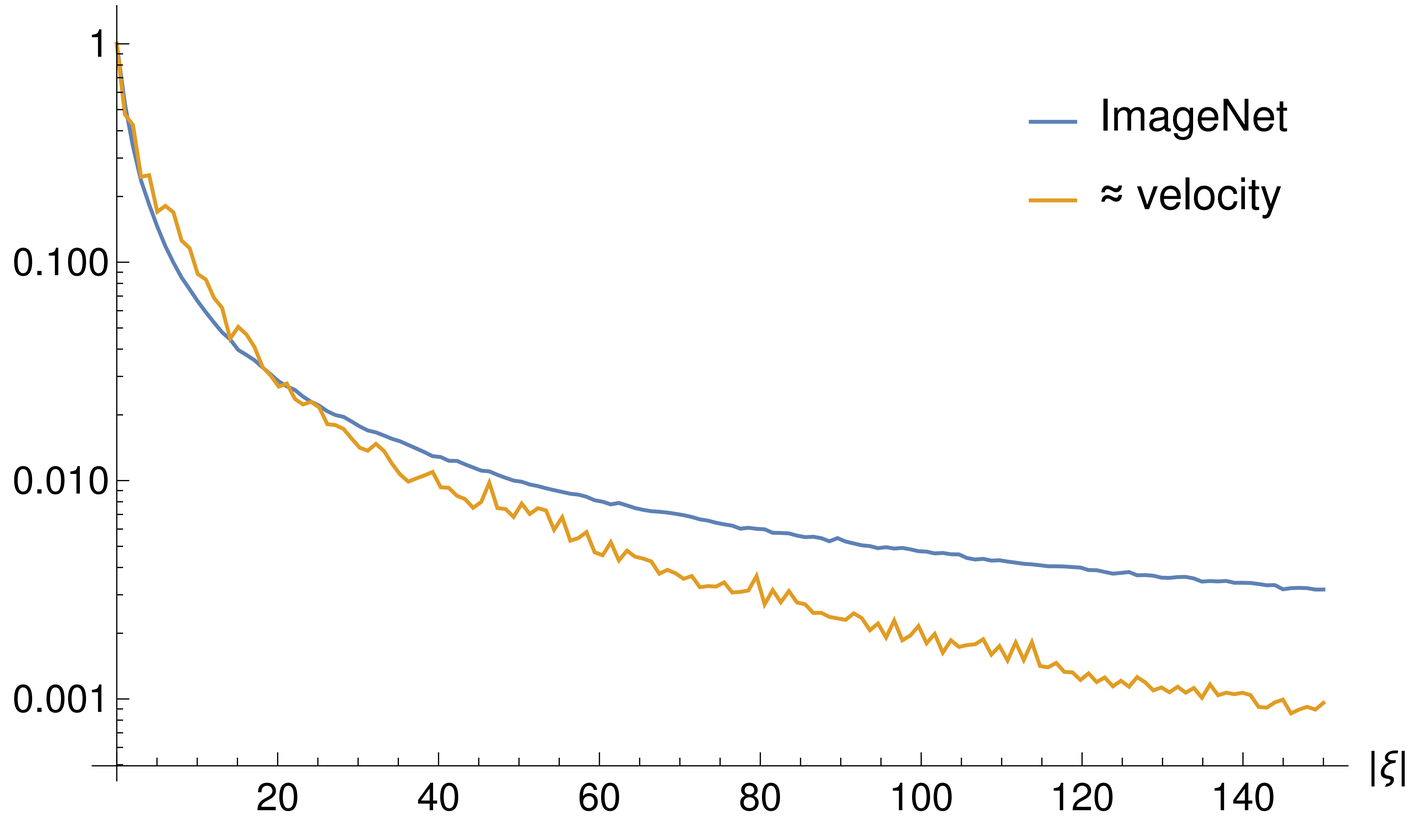
ERA5 versus ImageNet

stream function
velocity potential

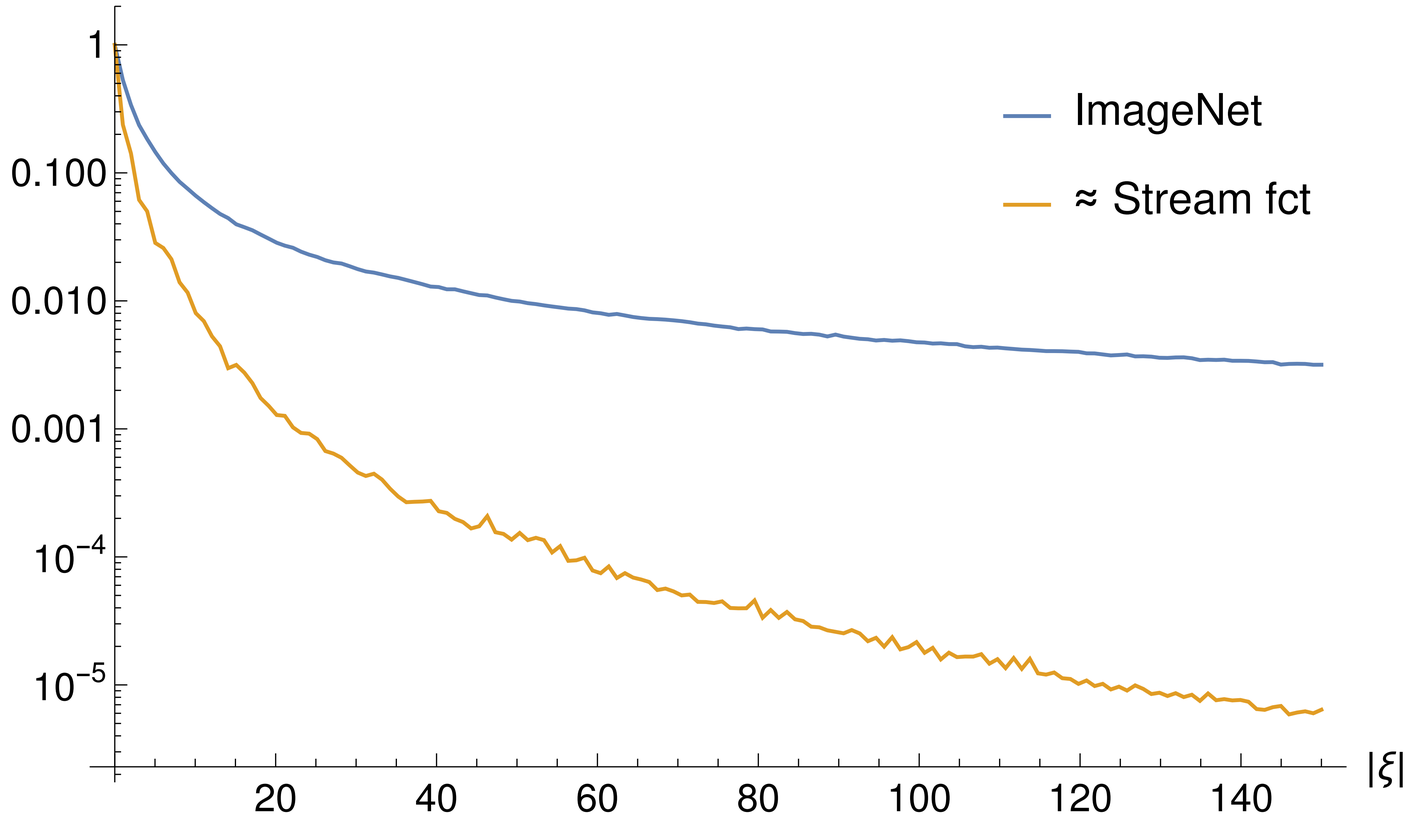
velocity
vector field

vorticity
divergence

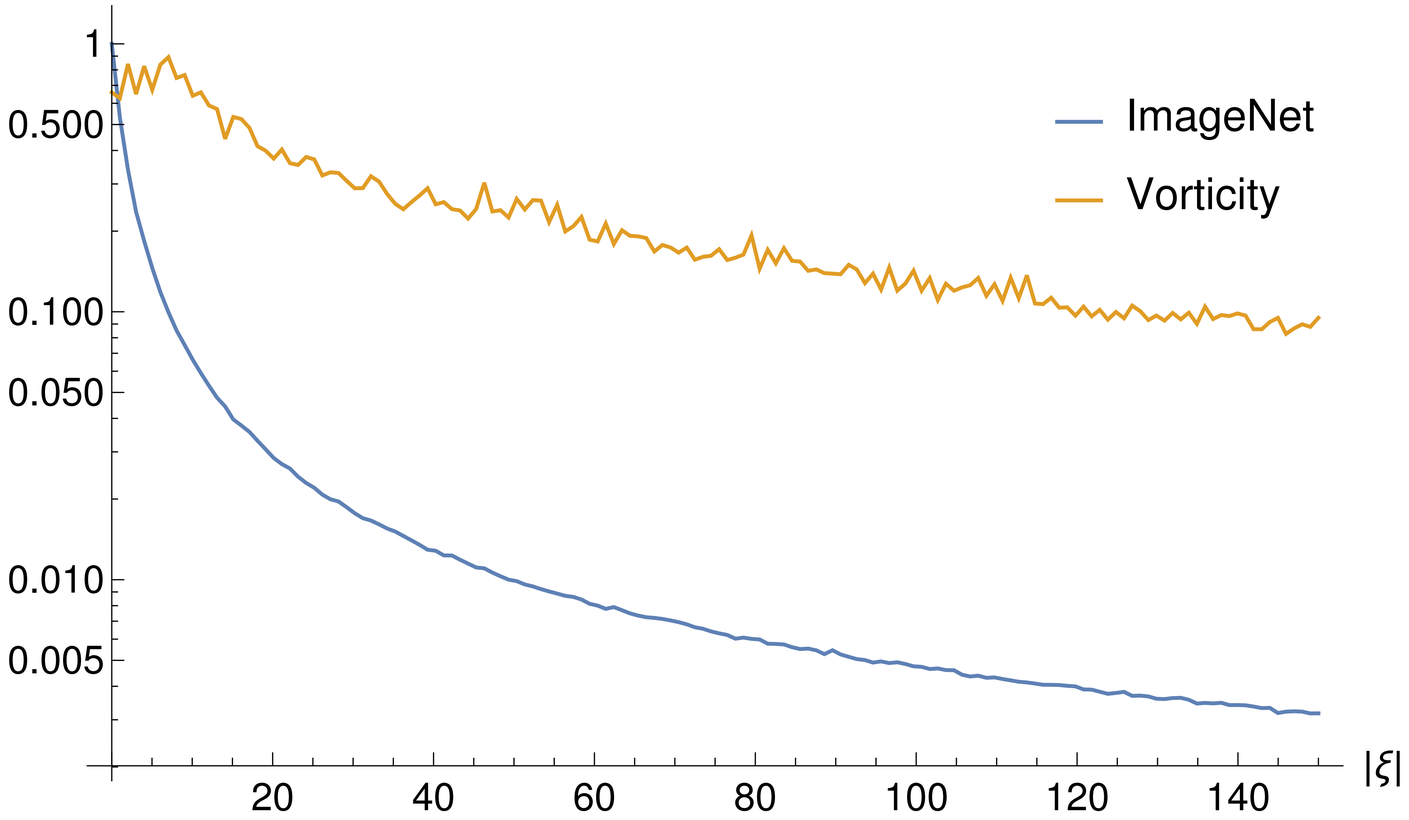
ERA5 versus ImageNet



ERA5 versus ImageNet

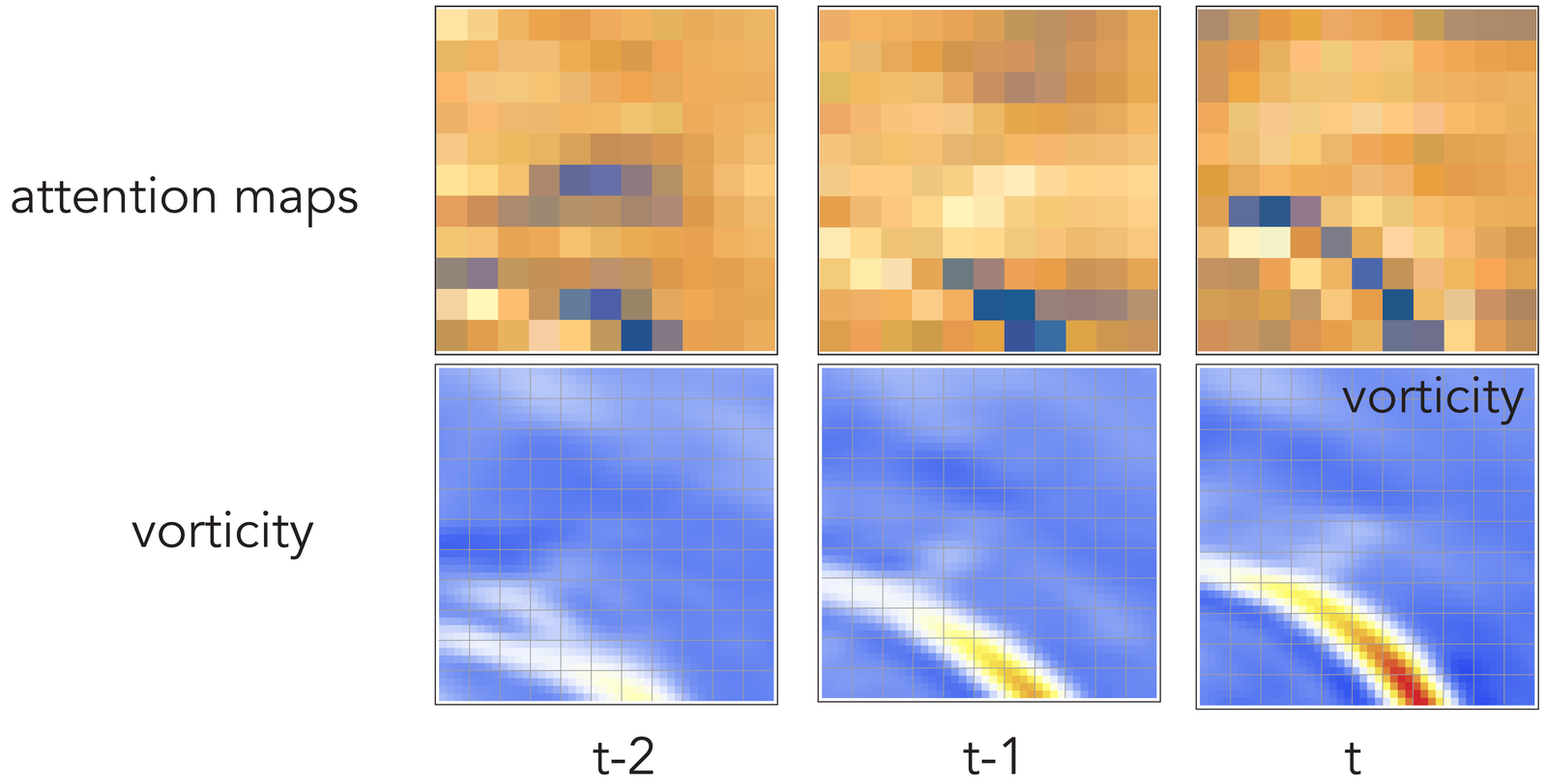


AtmoRep data



Statistical loss

- Attention maps:



Statistical loss

- Statistical loss:

$$\mathcal{L}_{\text{stats}} = \left| 1 - \int_{\mathrm{R}} \delta_y(x)\, G_{\tilde{\mu},\tilde{\sigma}}(x)\, \mathrm{d}x \right|^2$$

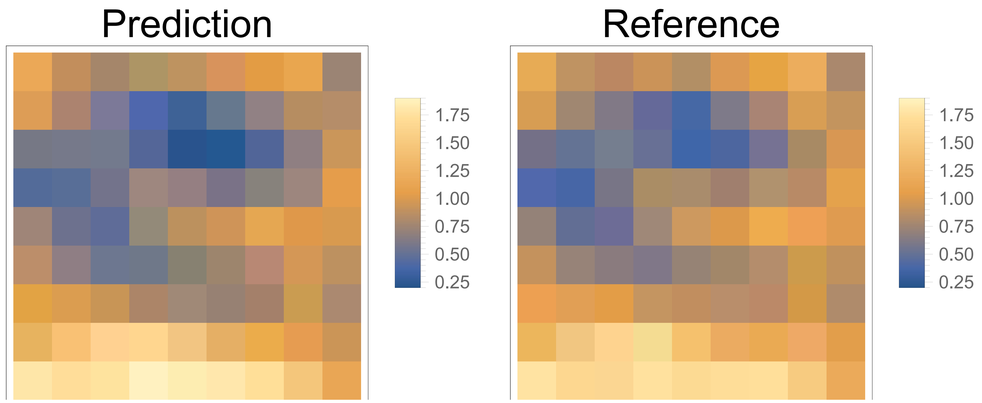
- CRPS:[1]

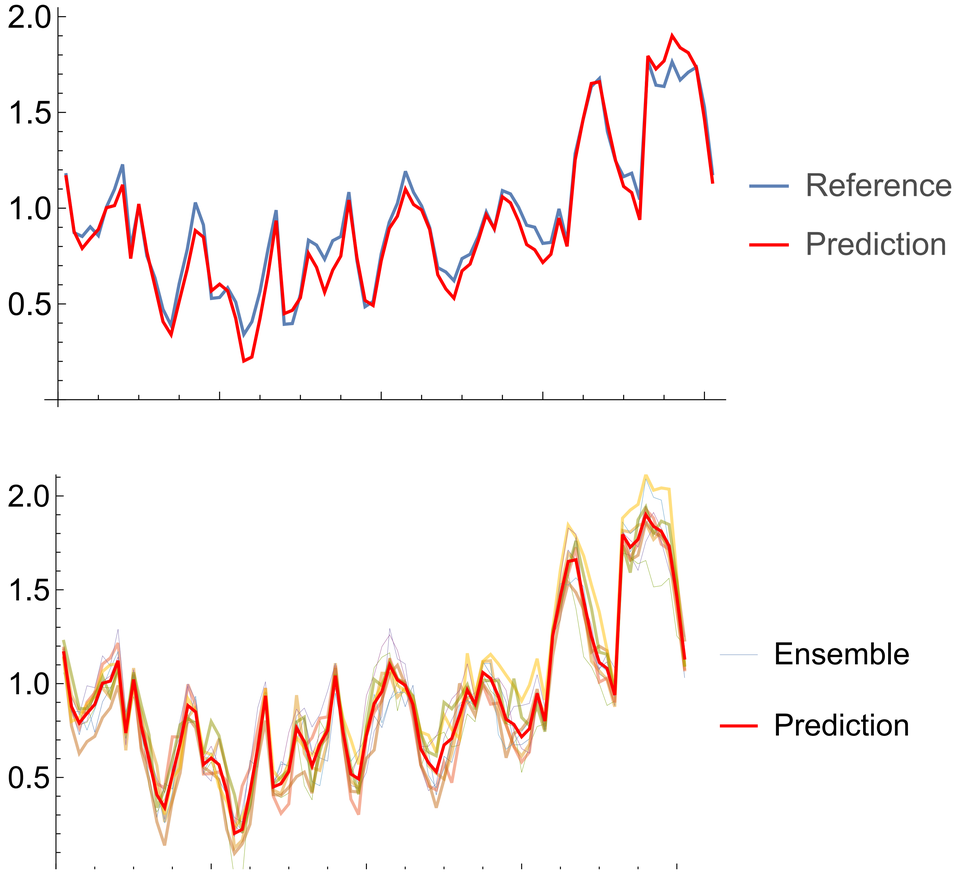
$$\mathcal{L}_{\text{CRPS}} = \int_{\mathrm{R}} \left| H_y(x)\, \mathrm{erf}_{\tilde{\mu},\tilde{\sigma}}(x) \right|^2 \mathrm{d}x$$

[1] S. Rasp and S. Lerch. Neural networks for postprocessing ensemble weather forecasts. Monthly Weather Review, 146(11):3885 – 3900, 2018.

Statistical loss

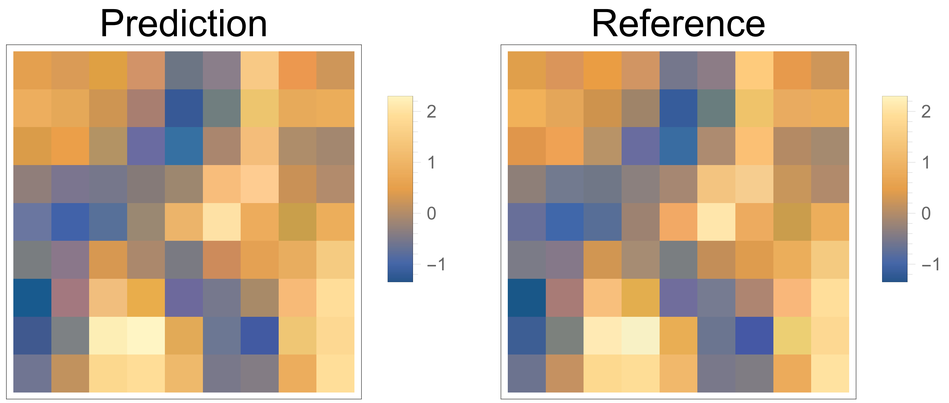
- Predictions:

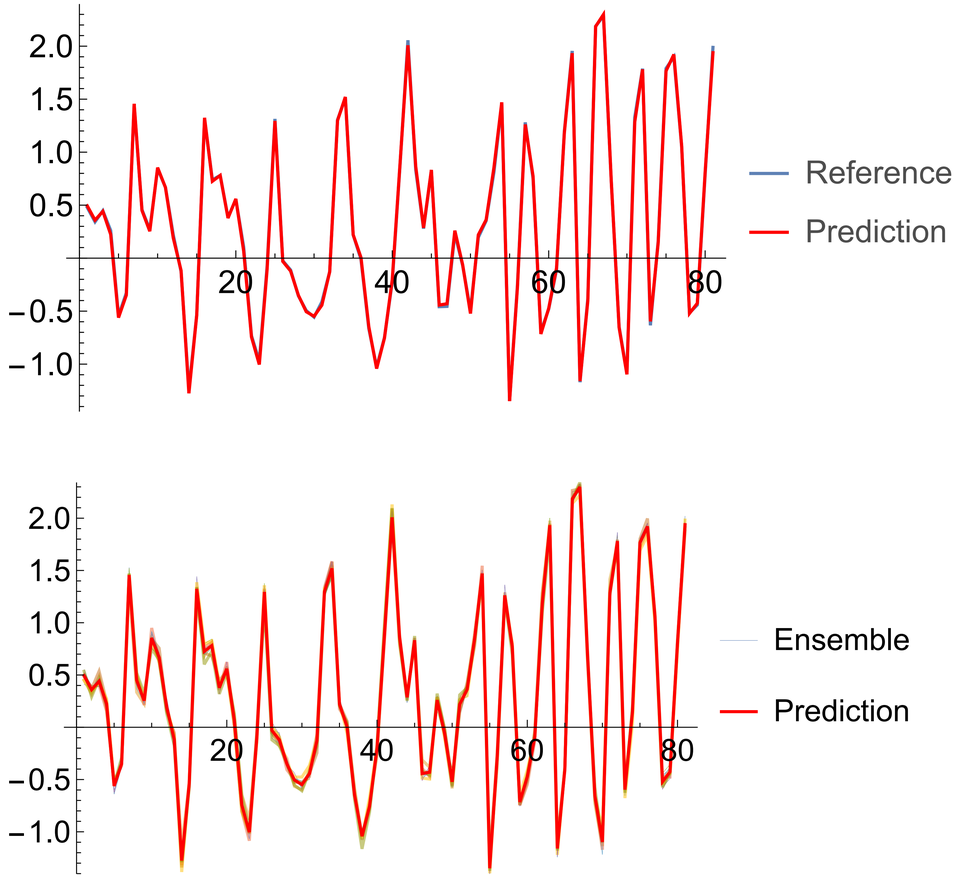




Statistical loss

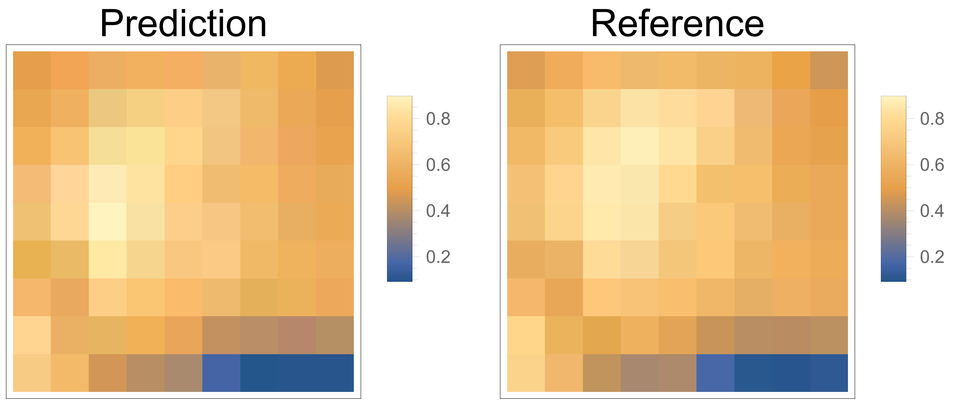
- Predictions:

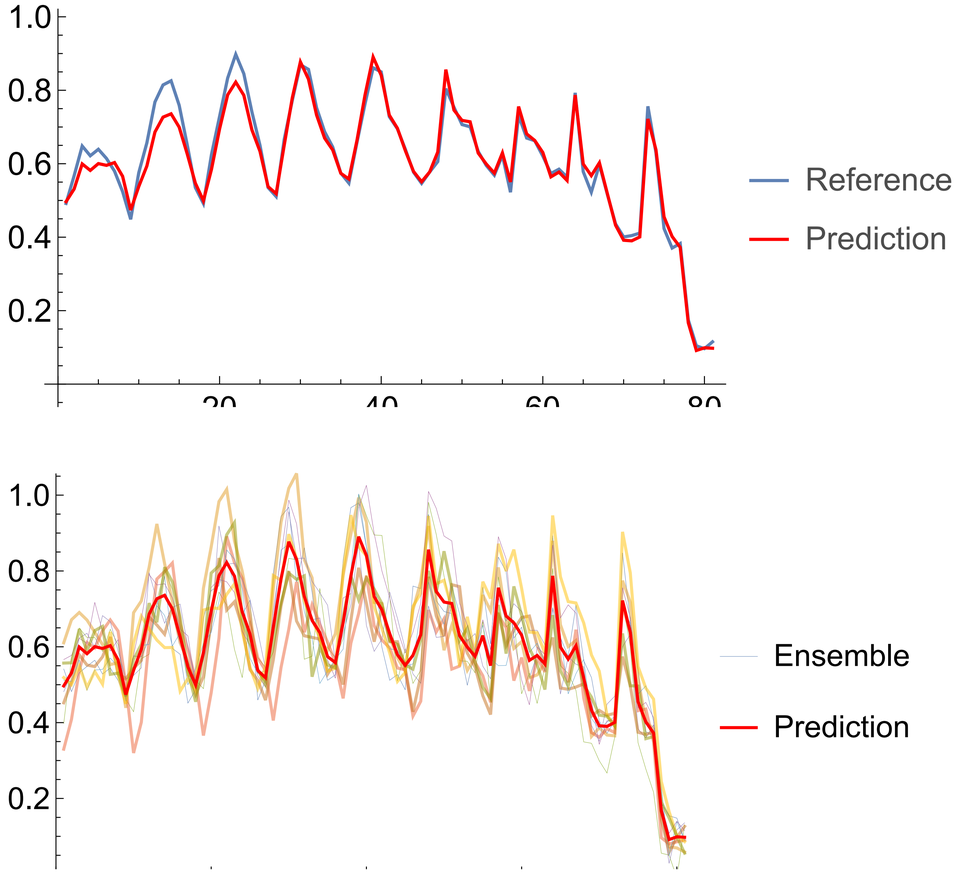




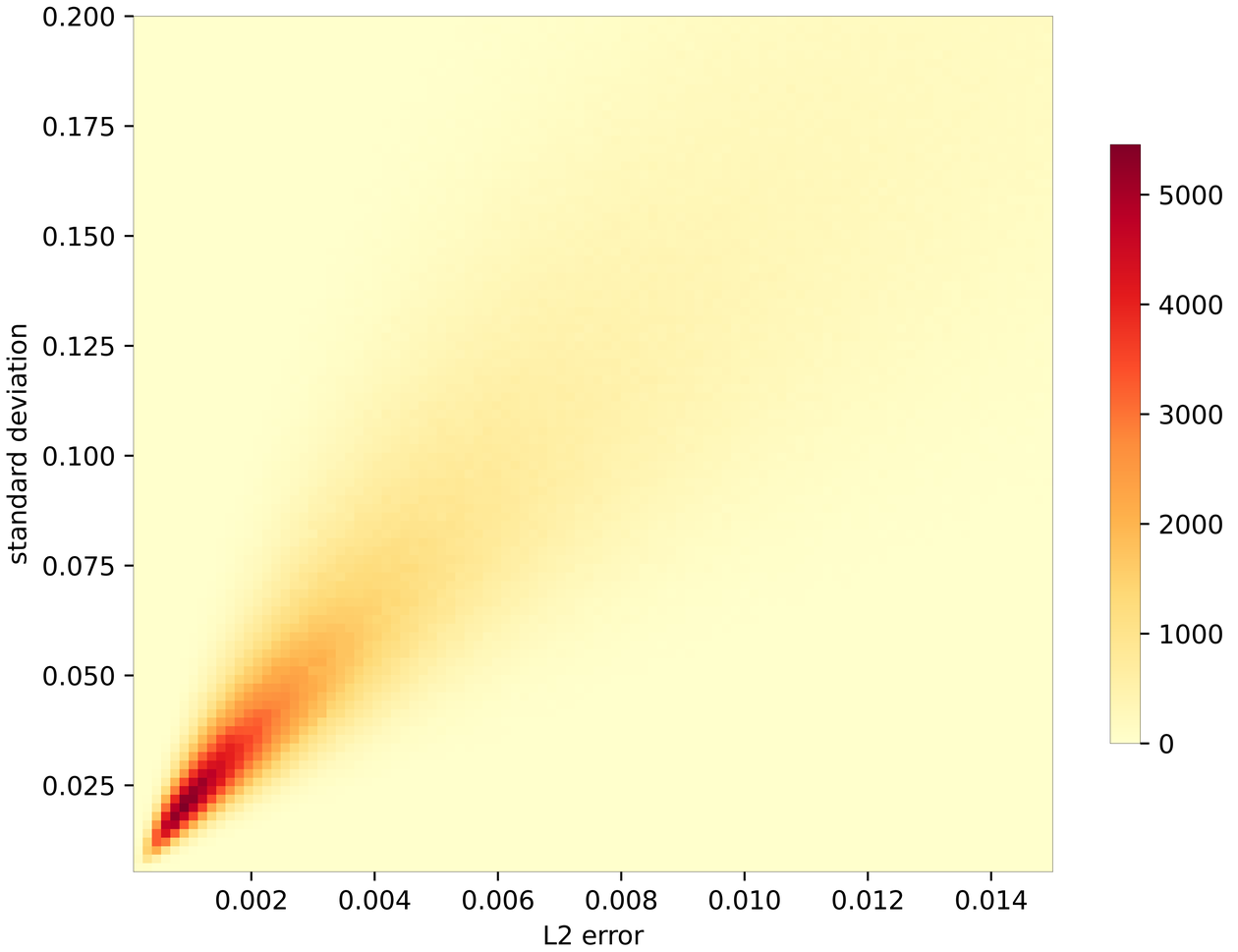
Statistical loss

- Predictions:



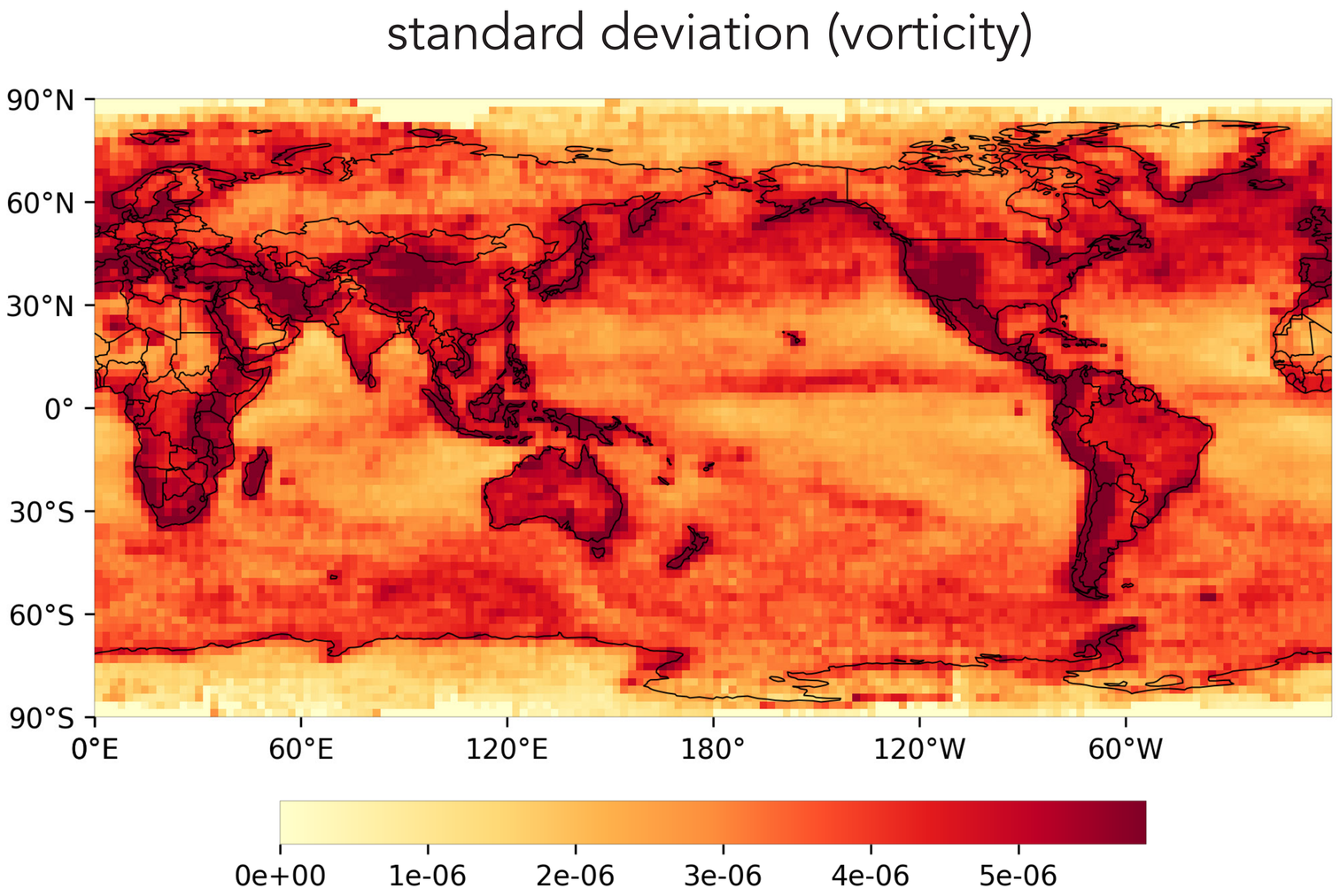


Statistical loss



2D Histogram of L_2 error vs. std. dev. (temperature)

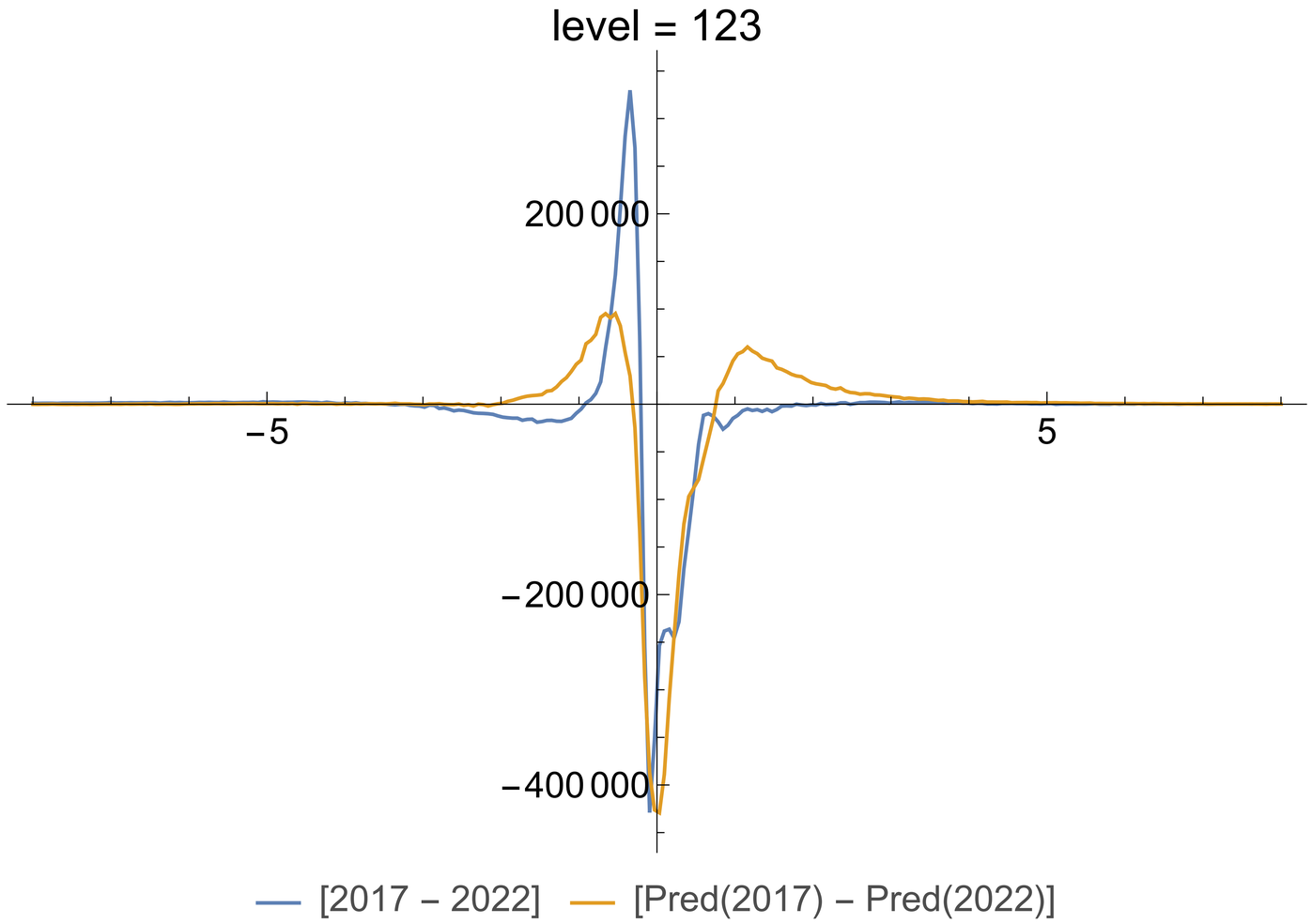
Statistical loss



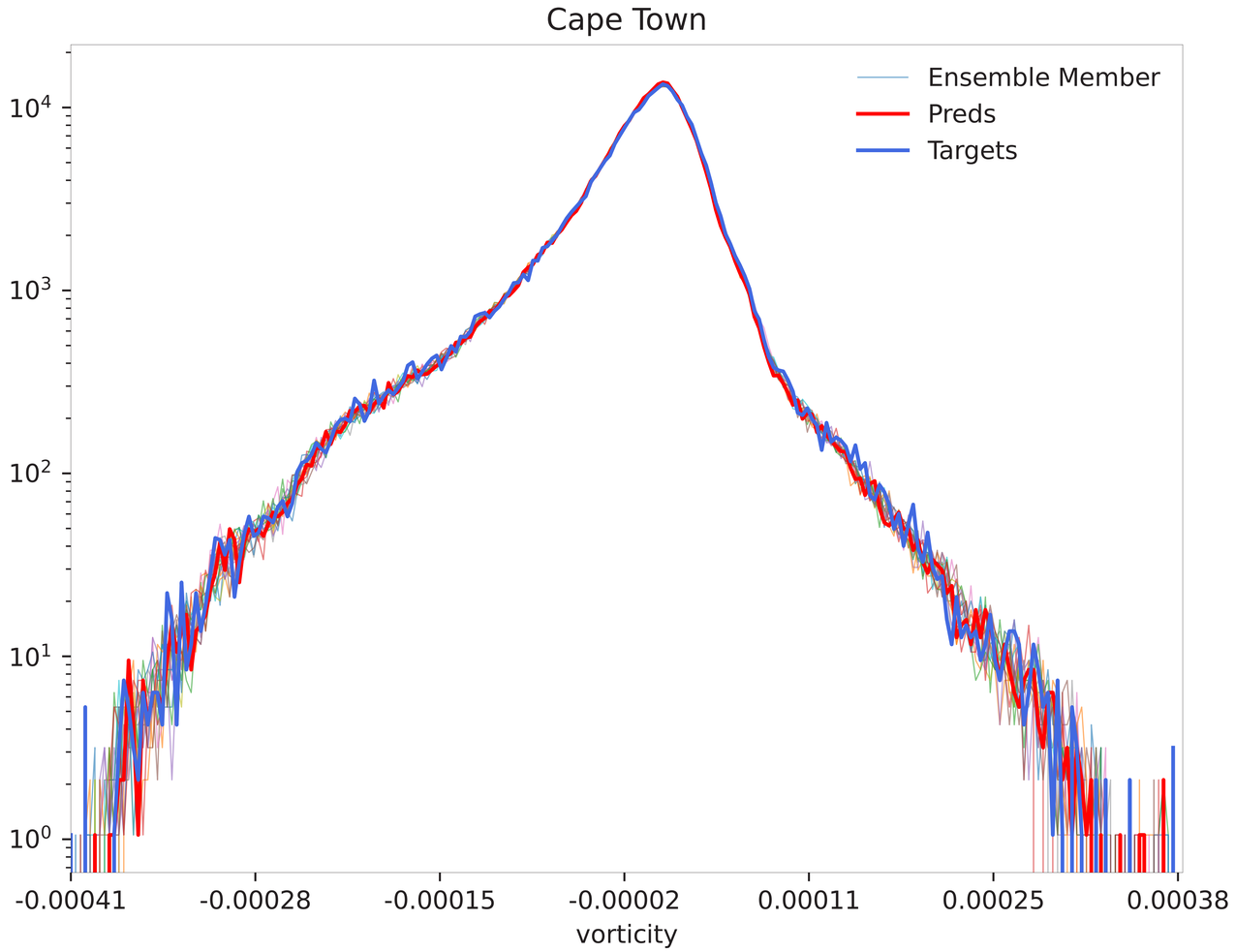
Some observations on training

- No overfitting
  - No difference between train and test set even with few months and limited spatial domain
- Different fields behave rather differently
  - Masking ratios have to be chosen differently
  - Statistical loss performs differently
- MSE loss accross fields not comparable and misleading

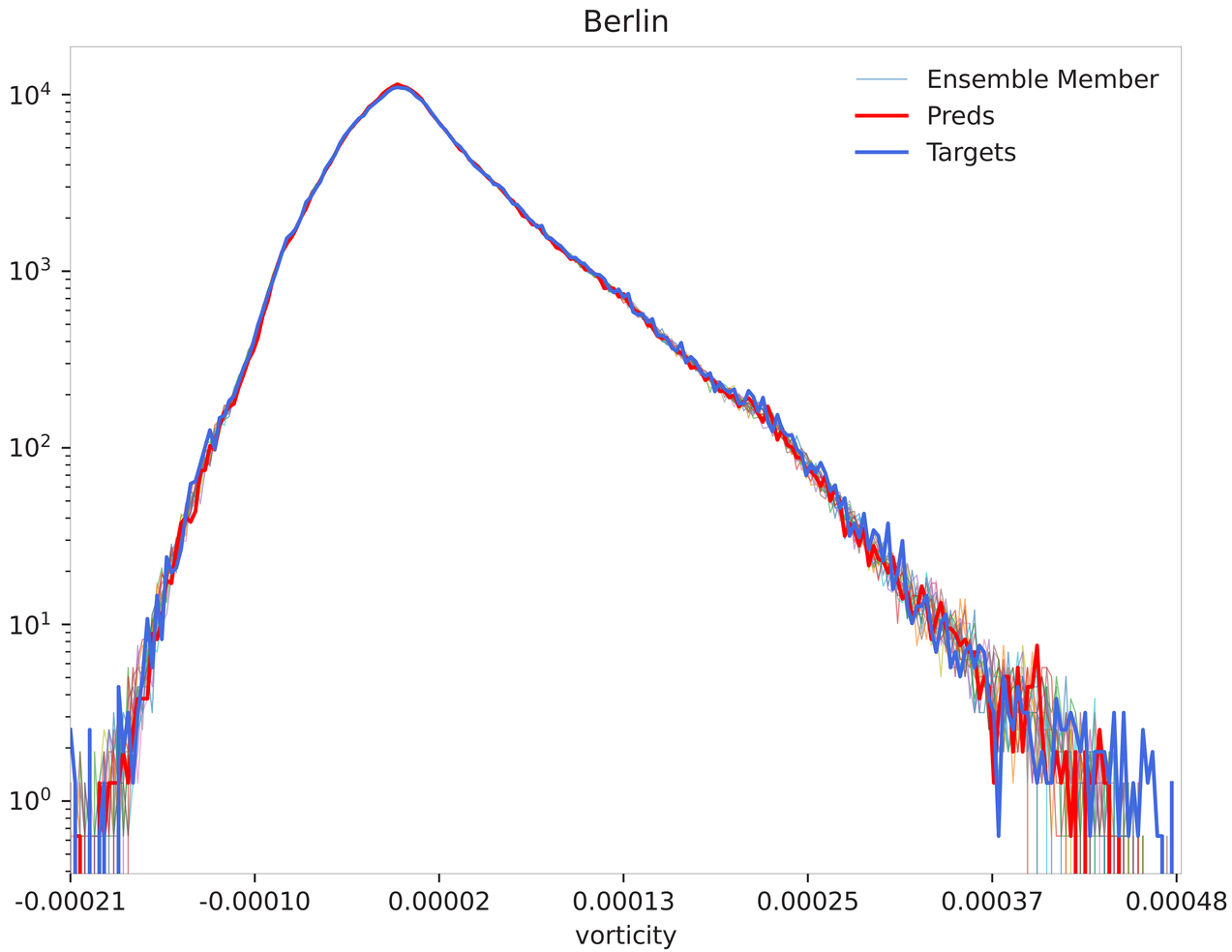
Counterfactuals: extrapolation



Pre-training results



Pre-training results



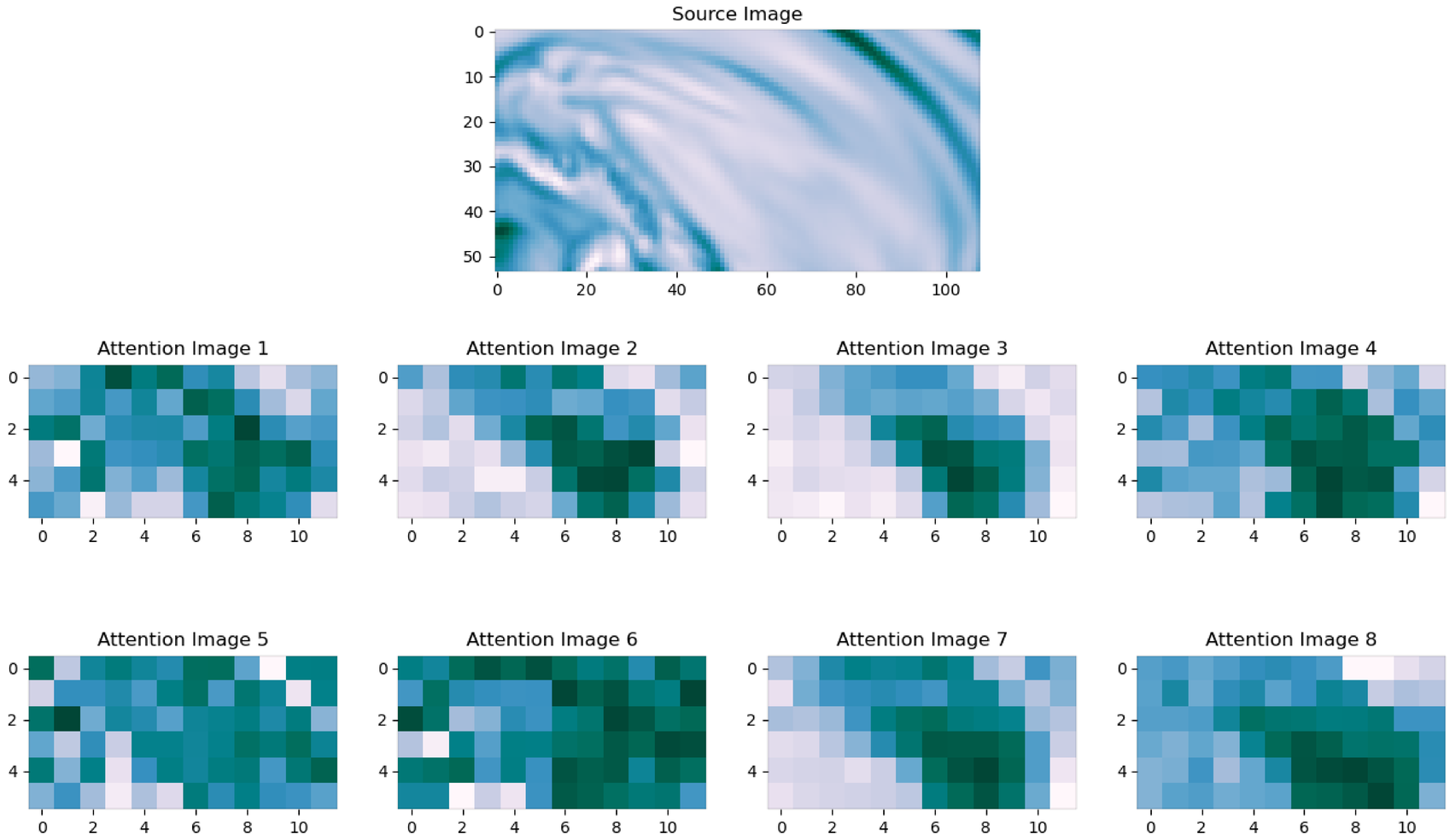
Training on observations

- Neural networks are models that work well on heterogenous and noisy data
- Fine-tune/bias correct a pre-trained model with observations (instead of training from scratch)

Can one continuous update a model with observations?

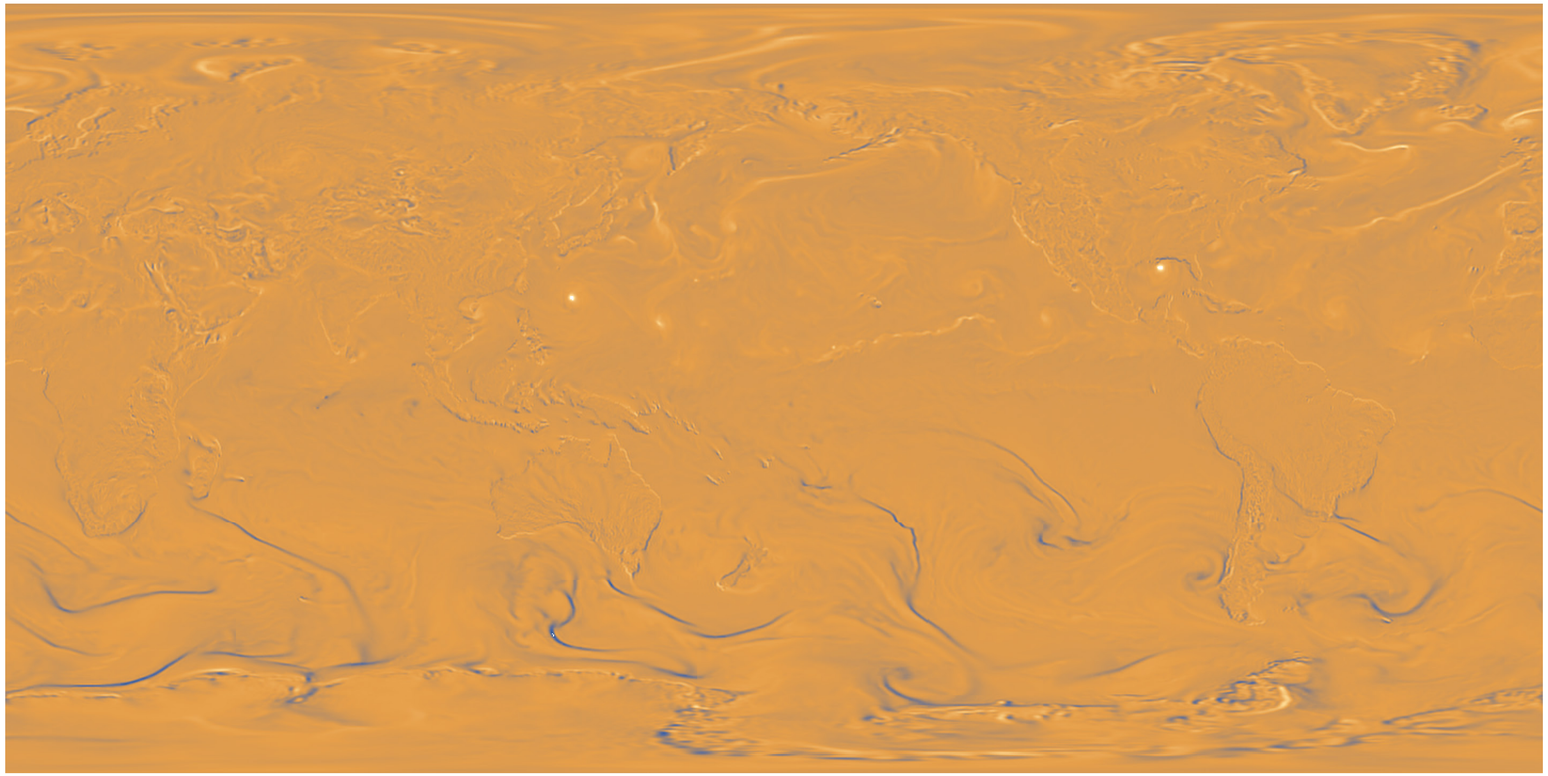
How to handle and propagate uncertainties?

Physics of trained network



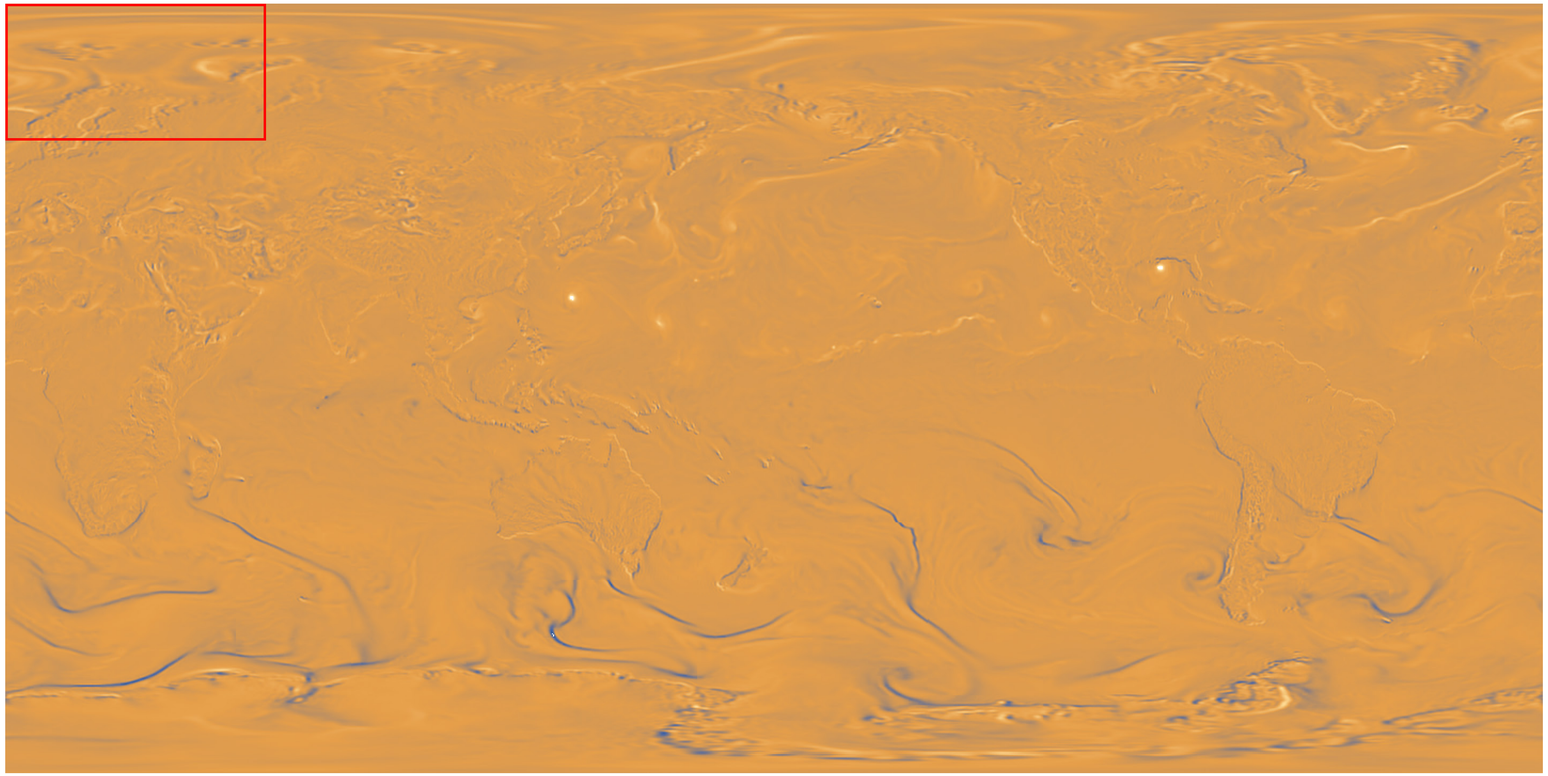
Medium range forecasting

- How to do global forecasts with a local model?



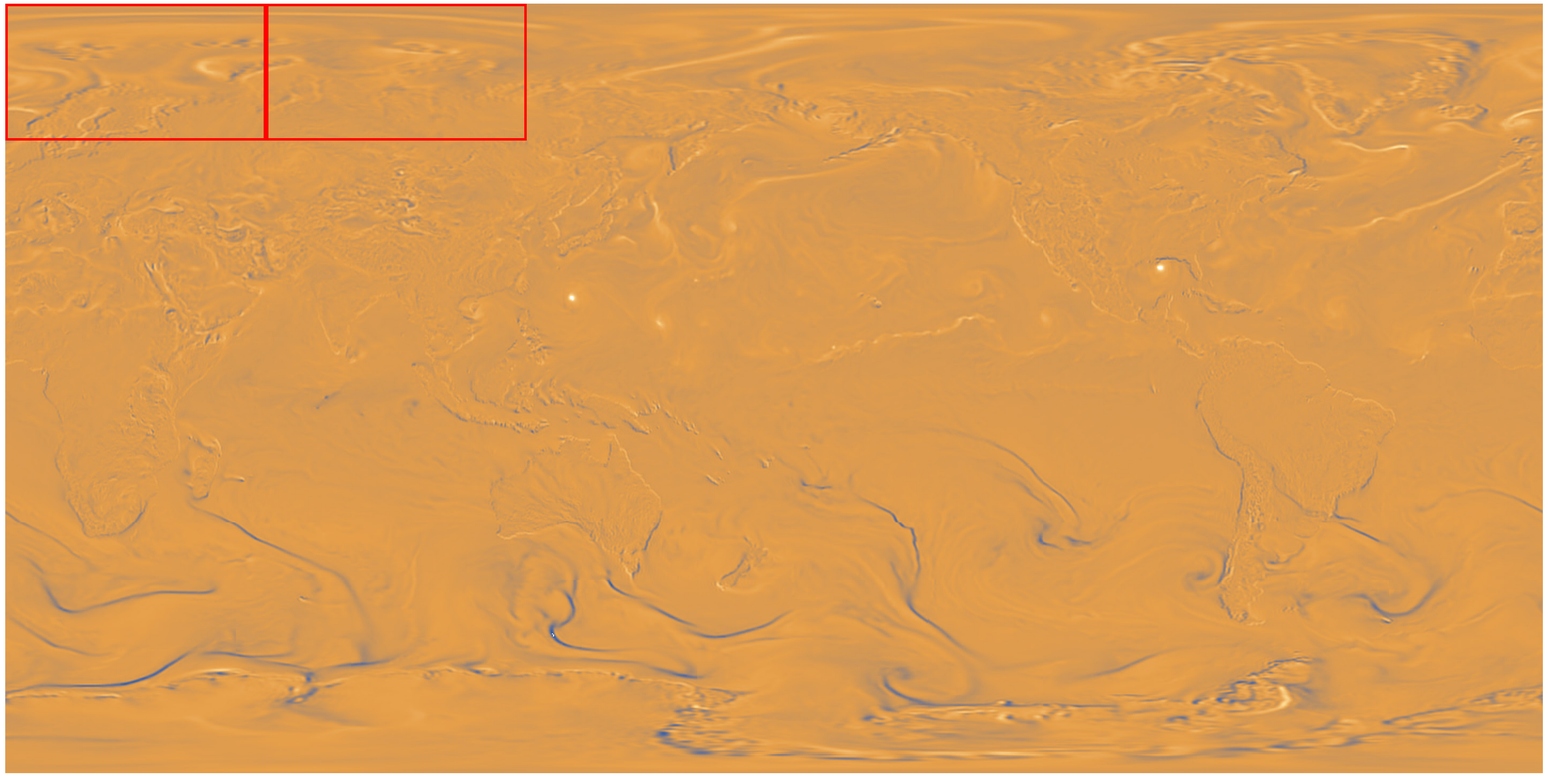
Medium range forecasting

- How to do global forecasts with a local model?



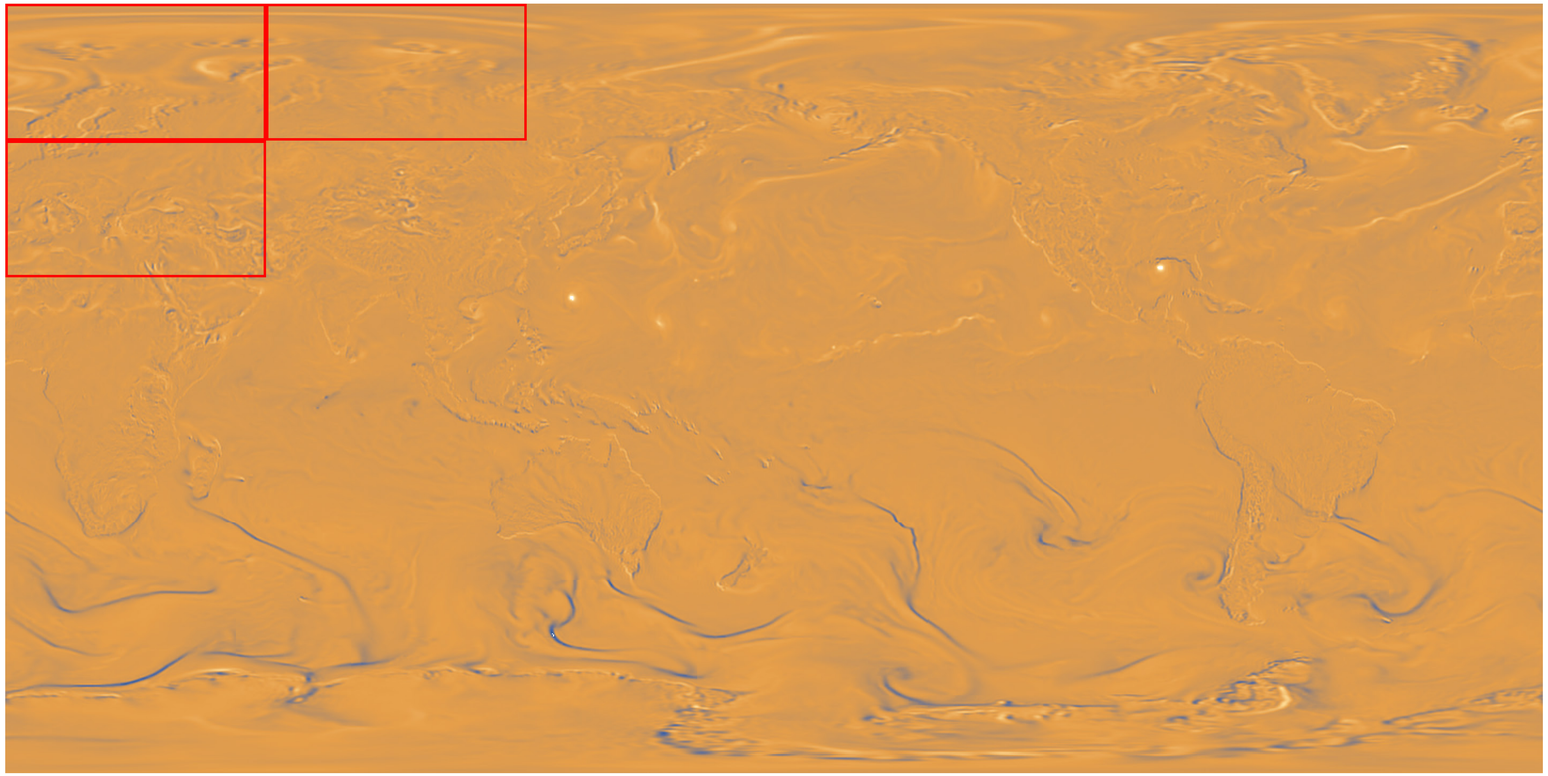
Medium range forecasting

- How to do global forecasts with a local model?



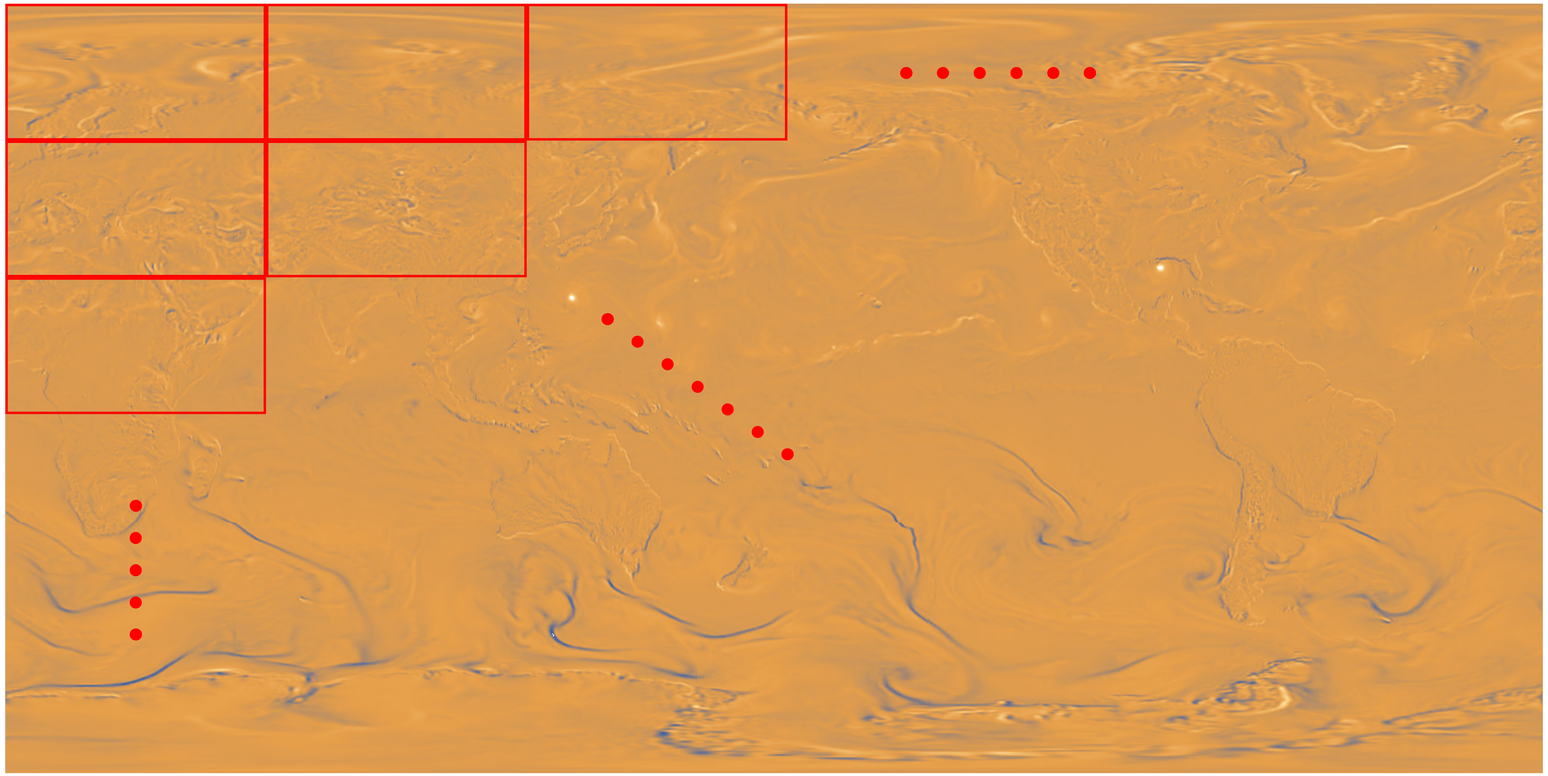
Medium range forecasting

- How to do global forecasts with a local model?



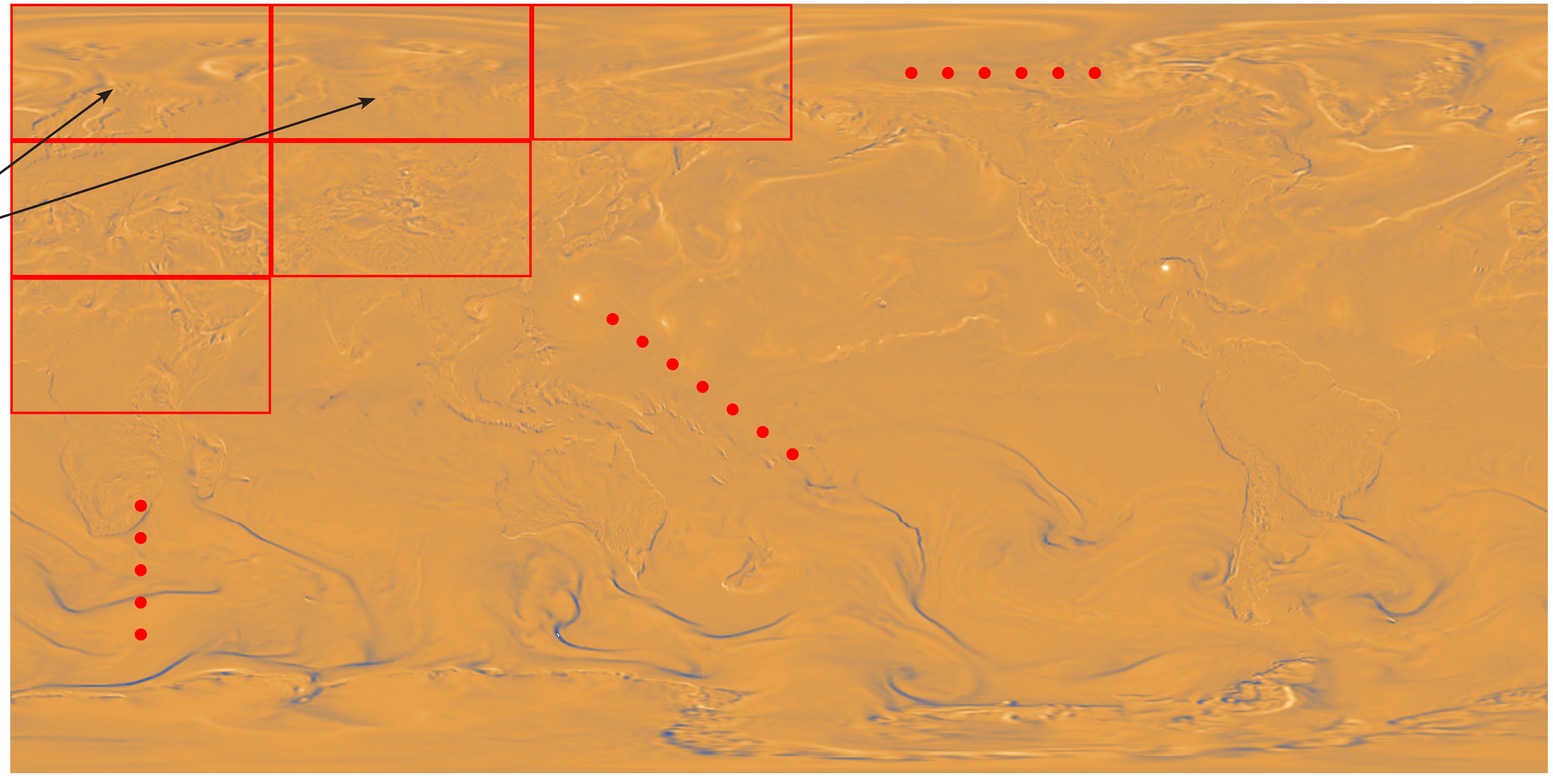
Medium range forecasting

- How to do global forecasts with a local model?



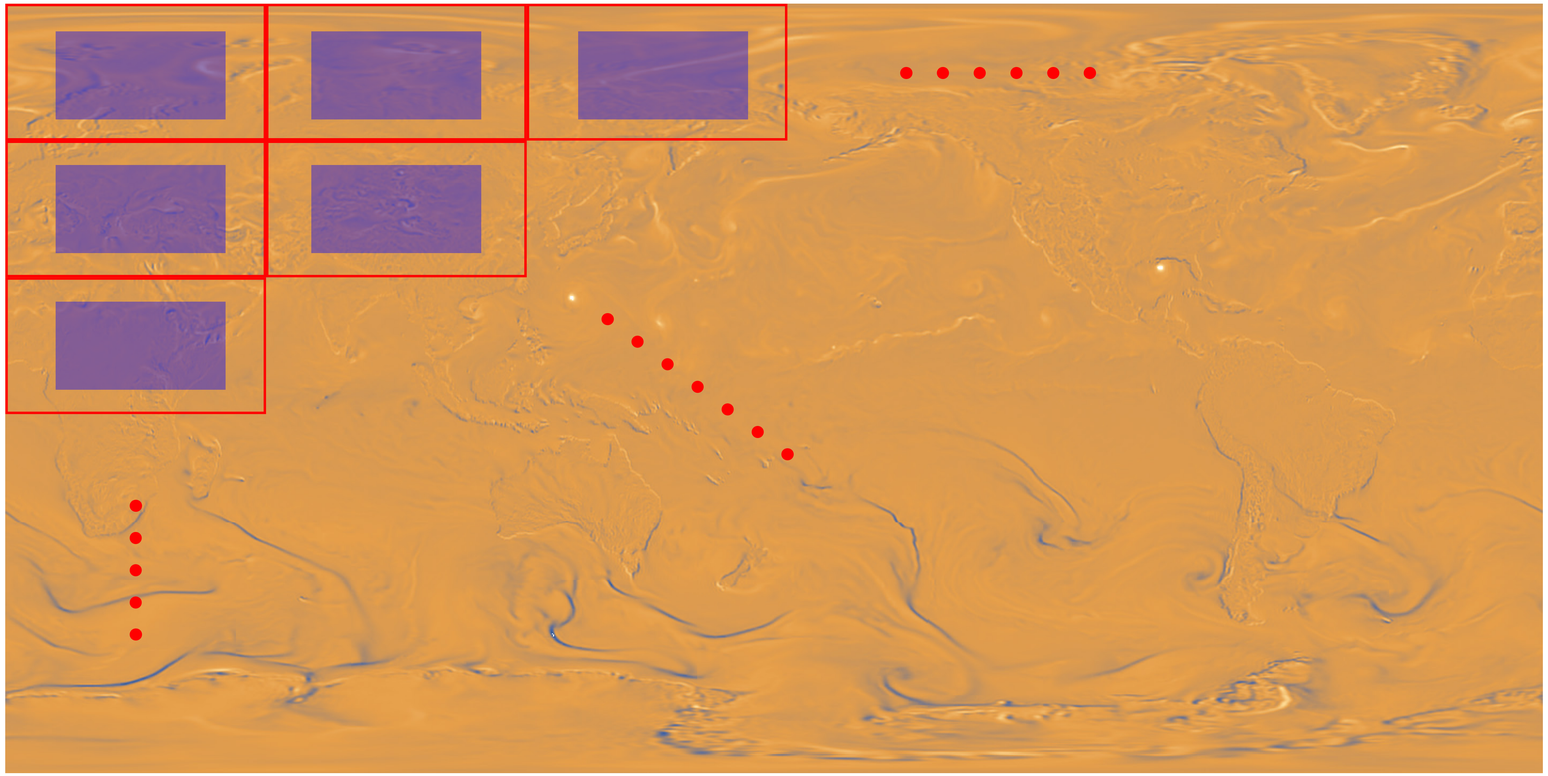
Medium range forecasting

- How to do global forecasts with a local model?



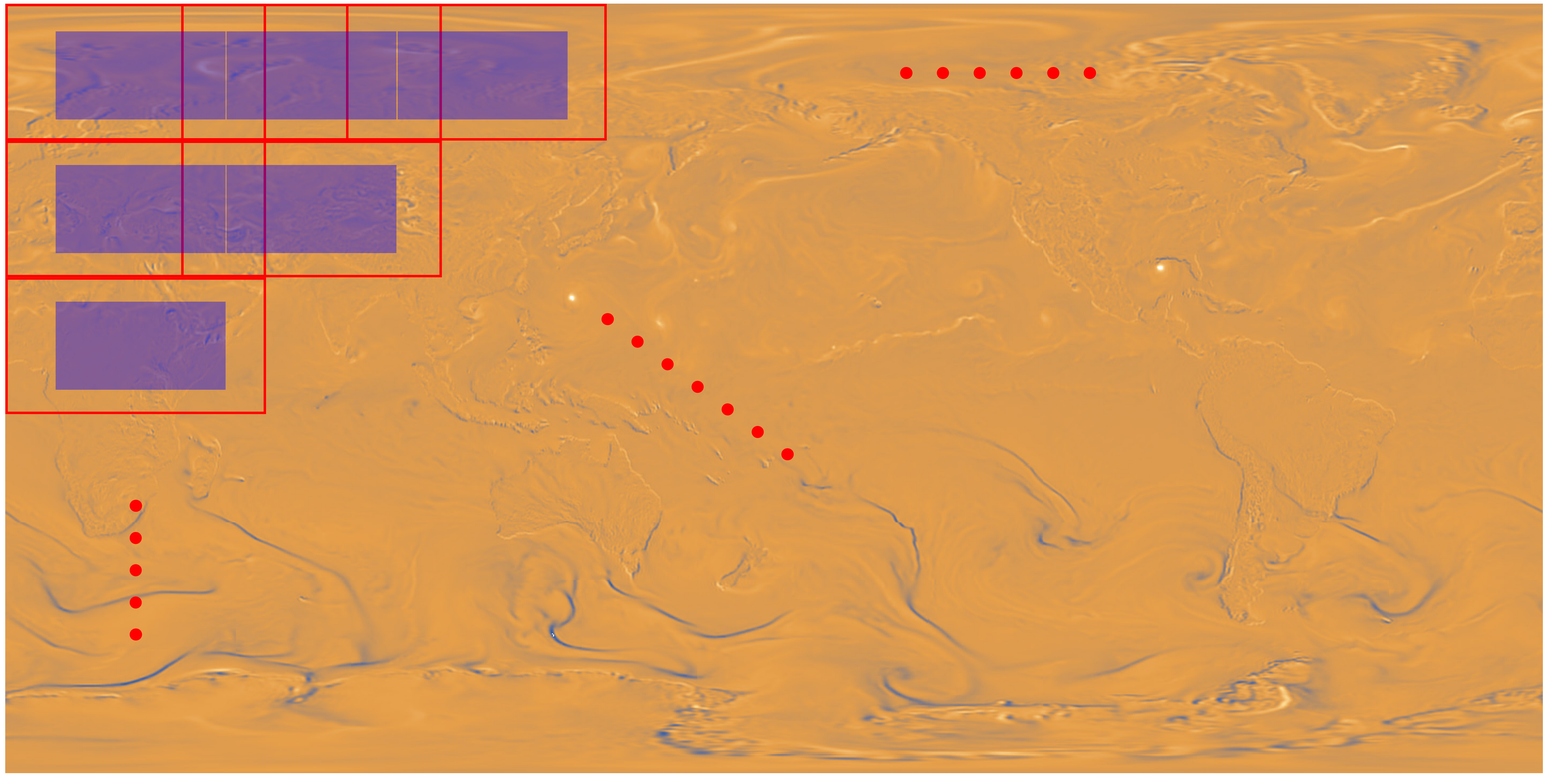
Medium range forecasting

- How to do global forecasts with a local model?



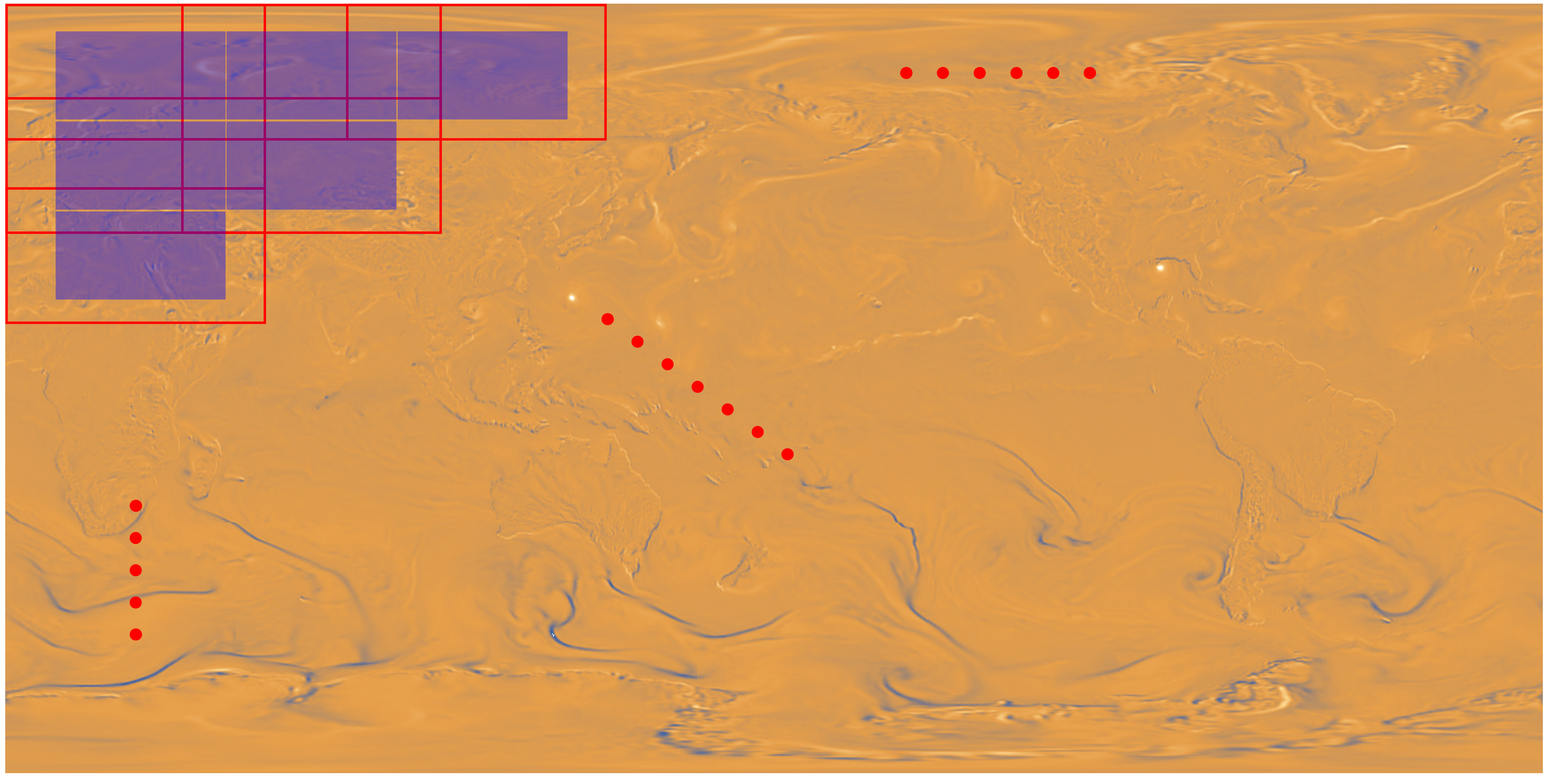
Medium range forecasting

- How to do global forecasts with a local model?



Medium range forecasting

- How to do global forecasts with a local model?



Multiformer: standard configuration

- Per token: $\alpha = \left(\mathrm{year}, \mathrm{day}, \mathrm{hour}, \mathrm{ml}, \theta, \phi, \mathrm{res}\right)$

Multiformer: standard configuration

- Per token: $\alpha = (\mathrm{year}, \mathrm{day}, \mathrm{hour}, \mathrm{ml}, \theta, \phi, \mathrm{res})$
- Neighhorhood: 36 h x 1350 km x 2700 km
  - 12 x 6 x 12 tokens with 3 x 9 x 9 grid points

Multiformer: standard configuration

- Per token: $\alpha = (\mathrm{year}, \mathrm{day}, \mathrm{hour}, \mathrm{ml}, \theta, \phi, \mathrm{res})$
- Neighhorhood: 36 h x 1350 km x 2700 km
  - 12 x 6 x 12 tokens with 3 x 9 x 9 grid points
- Fields: vorticity, divergence, vertical velocity, temperature, specific humidity, total precipitation, orography
- Vertical model levels: 96, 105, 114, 123, 137

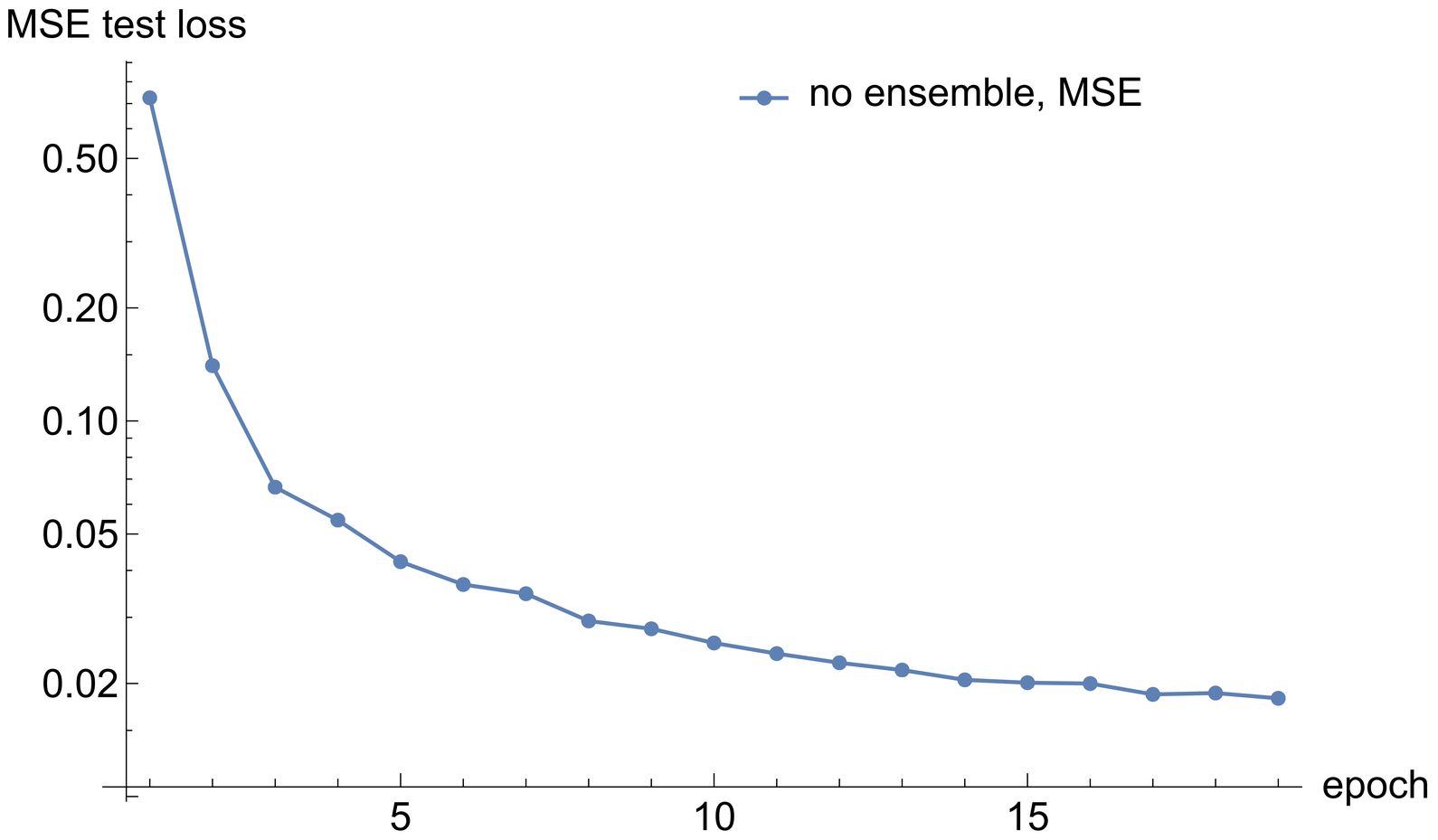
Multiformer: standard configuration

- Per token: $\alpha = (\mathrm{year}, \mathrm{day}, \mathrm{hour}, \mathrm{ml}, \theta, \phi, \mathrm{res})$
- Neighhorhood: 36 h x 1350 km x 2700 km
  - 12 x 6 x 12 tokens with 3 x 9 x 9 grid points
- Fields: vorticity, divergence, vertical velocity, temperature, specific humidity, total precipitation, orography
- Vertical model levels: 96, 105, 114, 123, 137
- Depth=20 (encoder: 10, decoder: 10) x 2048 embedding

Multiformer: standard configuration

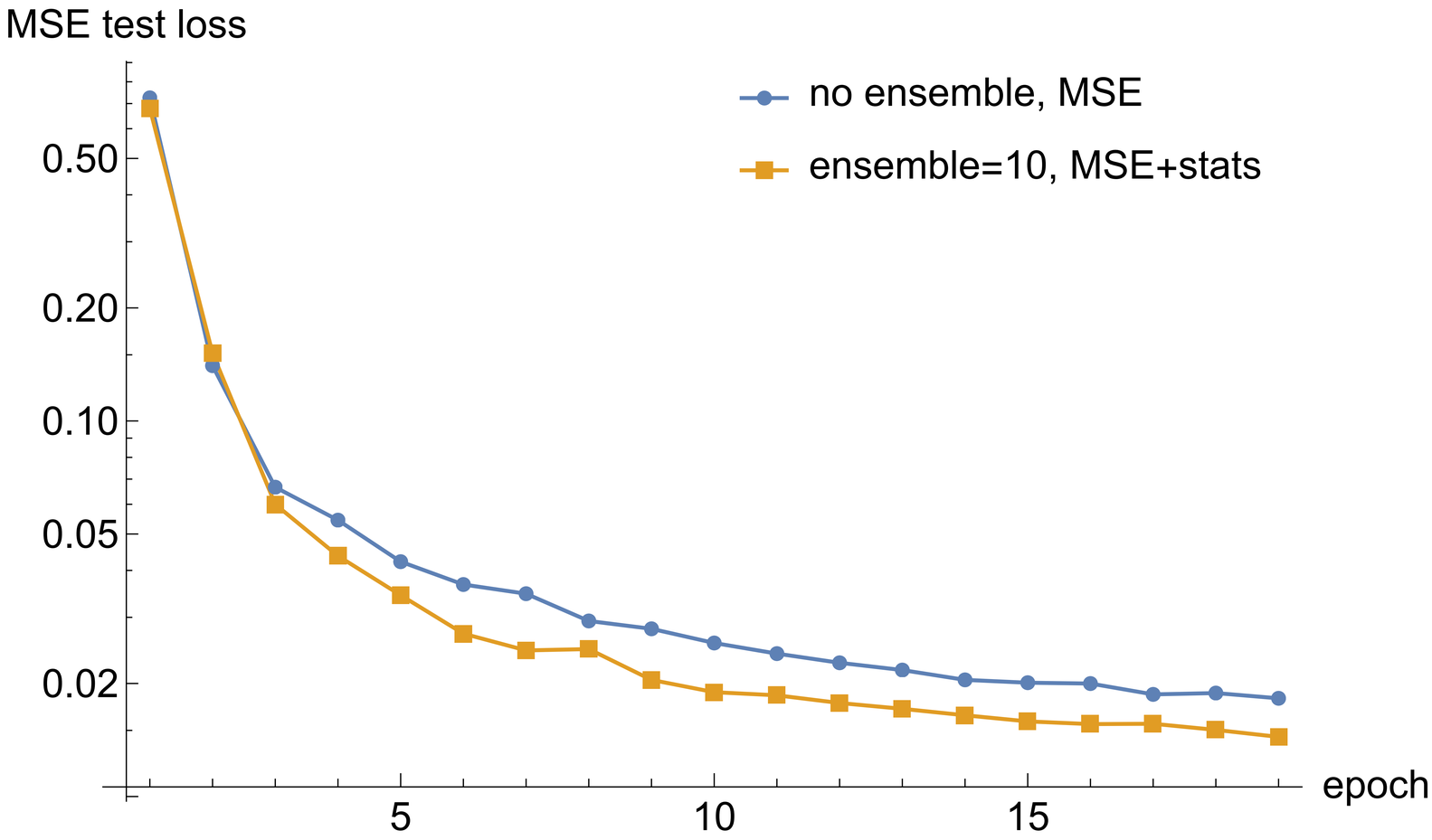
- Per token: $\alpha = (\mathrm{year}, \mathrm{day}, \mathrm{hour}, \mathrm{ml}, \theta, \phi, \mathrm{res})$
- Neighhorhood: 36 h x 1350 km x 2700 km
  - 12 x 6 x 12 tokens with 3 x 9 x 9 grid points
- Fields: vorticity, divergence, vertical velocity, temperature, specific humidity, total precipitation, orography
- Vertical model levels: 96, 105, 114, 123, 137
- Depth=20 (encoder: 10, decoder: 10) x 2048 embedding
- Total number of parameters: 3.5 billion

Statistical loss



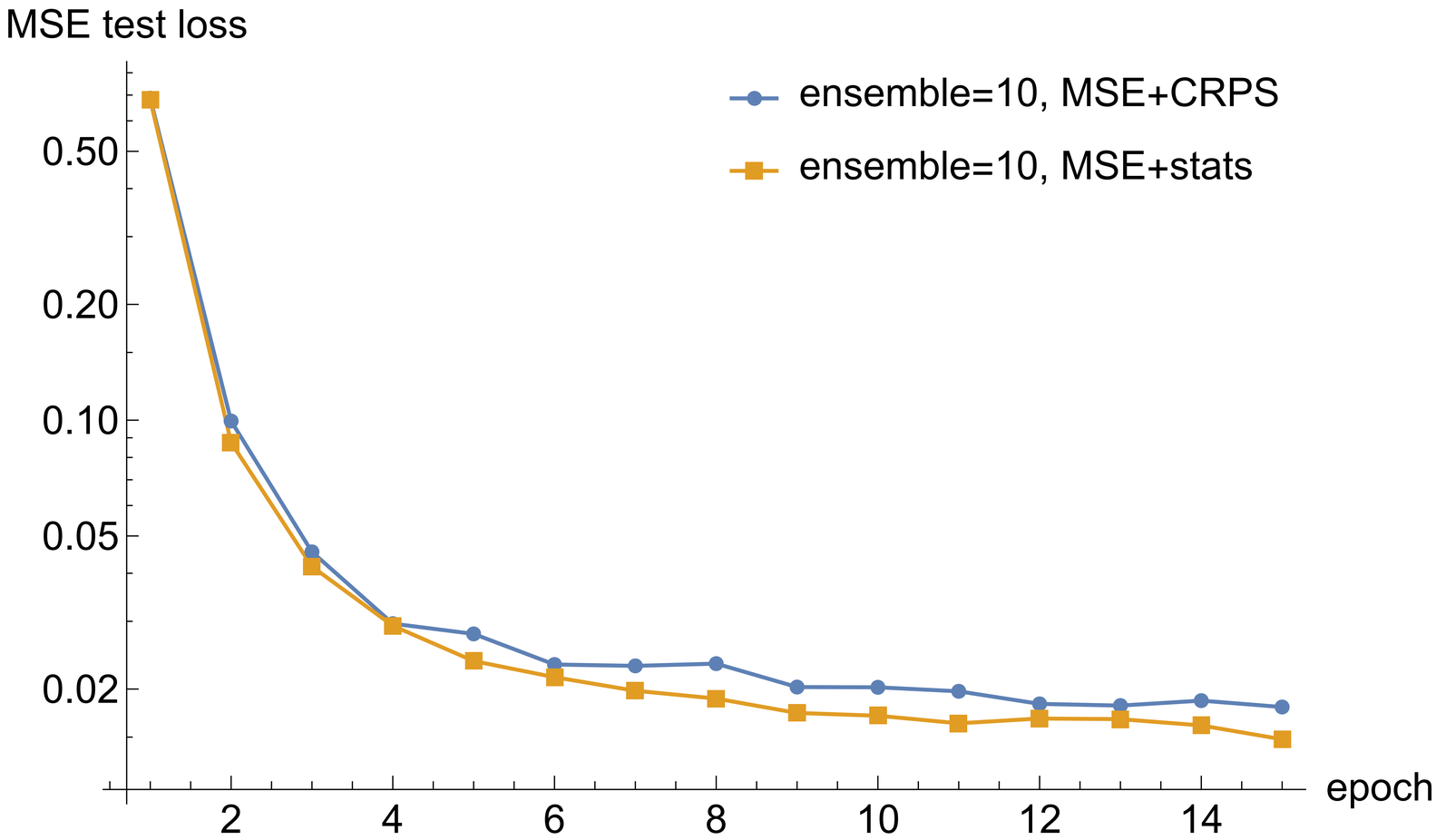
vorticity,
ml=137

Statistical loss



vorticity,
ml=123

Statistical loss



vorticity,
ml=123

Zero-shot capabilities

Large language model: $p_\theta\left(y|x\right)$

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

```
1  sea otter => loutre de mer          ← example #1
```

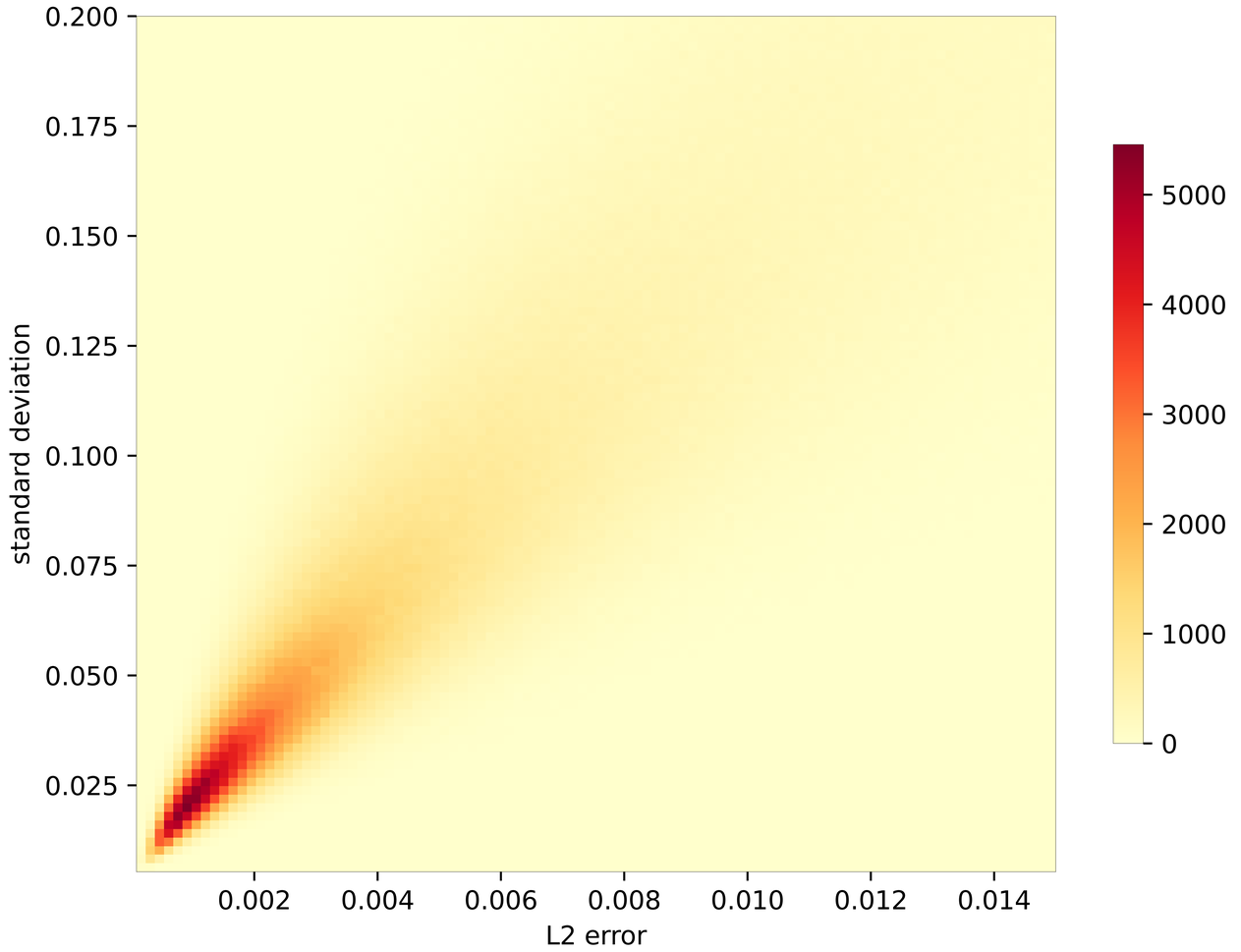
↓

gradient update

↓

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. *Language models are few-shot learners*. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

Statistical loss



2D Histogram of L_2 error vs. std.-dev. (temperature)

Pre-training results

- Pre-training of individual fields
  - More compute-efficient

Pre-training results

- Pre-training of individual fields
  - More compute-efficient
- Assembly of multiformer from pre-trained fields
  - Fields can be assembled as needed for application
  - Very little training time needed to “synchronize” pre-trained fields in assembled multiformer

Pre-training results