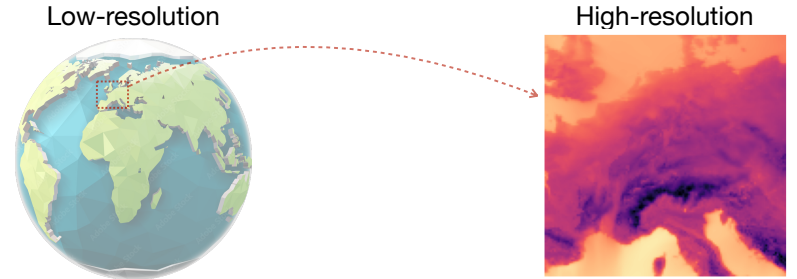
Generative Climate Modeling: Emulation and Dimension Reduction

Xinwei Shen

Department of Statistics, University of Washington

January 23, 2026

Climate models



- High computational cost for large ensemble size
- A small number of prescribed forcing scenarios

Statistical emulation of climate models

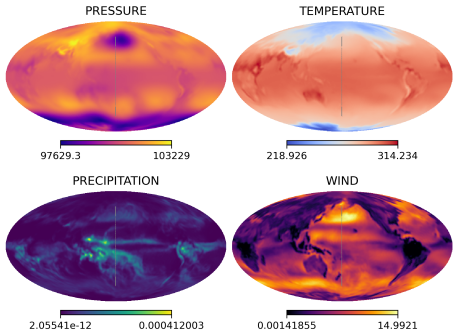
Motivation:

- Save computational cost for large ensembles
- Simulate across a continuous range of forcing pathways

Statistical emulators are not developed to replace physical climate models, but to serve as complements with benefits in computation, flexibility, etc.

Challenges in statistical emulation of climate models

- **High-dimensional**
  Eg, monthly pressure, temperature, precipitation, wind speed, each on a 180×360 grid



- **Distributional**: more comprehensive uncertainty quantification
  - single point predictions from a deterministic model ✗
  - large ensembles from a stochastic model ✓

Part I Distributional Learning

with Nicolai Meinshausen

Regression

Response $Y \in \mathbb{R}^d$; predictors $X \in \mathbb{R}^p$

Target: $P_{Y|X=x}$

- Deterministic regression

$$\hat{Y} = f(X)$$

  for conditional mean or median estimation ✗

- Distributional (stochastic) regression

$$\hat{Y} = g(X, \varepsilon)$$

  where $\varepsilon \sim P_\varepsilon$ pre-defined, for conditional distribution estimation ✓

Distributional learning via generative models

- Target: conditional distribution of $Y|X$
- Build a **generative model** to describe the distribution of $Y|X$:

$$Y = g(X, \varepsilon)$$

  where $\varepsilon \sim P_\varepsilon$ pre-defined and map $g : (x, \varepsilon) \mapsto y$ is often parametrized by neural networks.
- Goal: find g such that $g(x, \varepsilon) \sim P_{Y|X=x}$ for any x
- Sampling-based prediction: a model to sample from $P_{Y|X=x}$

Our distributional learning method *engression*

- Engression solution:[1]

$$\tilde{g} \in \operatorname*{argmin}_{g \in \mathcal{G}} \mathbb{E}_{(X,Y)\sim P}[-S(P_g(.|X), Y)]$$
$$= \operatorname*{argmin}_{g \in \mathcal{G}} \mathbb{E}\Big[\|Y - g(X,\varepsilon)\|_2 - \frac{1}{2}\|g(X,\varepsilon) - g(X,\varepsilon')\|_2\Big]$$

  where $P_g(.|x)$ is the distribution of $g(x,\varepsilon)$ and $\varepsilon, \varepsilon'$ are independent draws from $\mathcal{N}(0, I)$.

- **Proposition**: under correct model specification, we have for any $x \in \mathrm{supp}(P_X)$,

$$\tilde{g}(x,\varepsilon) \sim P_{Y|X=x}$$

  This gives us a simulator for $Y|X$, not just a point predictor.

[1] Energy score by Gneiting and Raftery, 2007

Sampling-based predictions

For a fixed test point x, e.g., a certain day or forcing condition

Goal: to predict about Y given $X = x$.

1. Draw a sample of ε, i.e., $\varepsilon_1, \ldots, \varepsilon_m$;
2. **Ensemble**: $\{\tilde{g}(x, \varepsilon_i),\ i = 1, \ldots, m\}$ is a sample of the estimated distribution $Y|X = x$;
3. **Estimation of climate-relevant quantities**:
   - conditional mean: $\mathbb{E}[Y|X = x] \approx \hat{\mathbb{E}}_\varepsilon[\tilde{g}(x, \varepsilon)]$
   - conditional α-quantiles: $Q_\alpha(Y|X = x) \approx \hat{Q}_\alpha(\tilde{g}(x, \varepsilon))$
   - correlations: $\mathrm{Corr}(Y_i, Y_j|X = x) \approx \hat{\mathrm{Corr}}([\tilde{g}(x, \varepsilon)]_i, [\tilde{g}(x, \varepsilon)]_j)$

Our R and Python packages[1]

R: `install.packages("engression")`

```
> library(engression)                                                    ## load engression package
> engressor = engression(X, Y)                                           ## fit an engression model
> predict(engressor, Xtest, type="mean")                                 ## mean prediction
> predict(engressor, Xtest, type="quantile", quantiles=c(0.1, 0.5, 0.9)) ## quantile prediction
> predict(engressor, Xtest, type="sample", nsample=100)                  ## sampling
```

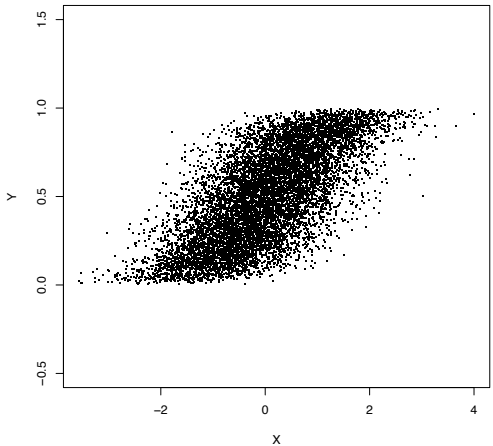
Python: `pip install engression`

```
> from engression import engression                                      ## load engression package
> engressor = engression(X, Y)                                           ## fit an engression model
> engressor.predict(Xtest, target="mean")                                ## mean prediction
> engressor.predict(Xtest, target=[0.025, 0.5, 0.975])                   ## quantile prediction
> engressor.sample(Xtest, sample_size=100)                               ## sampling
```

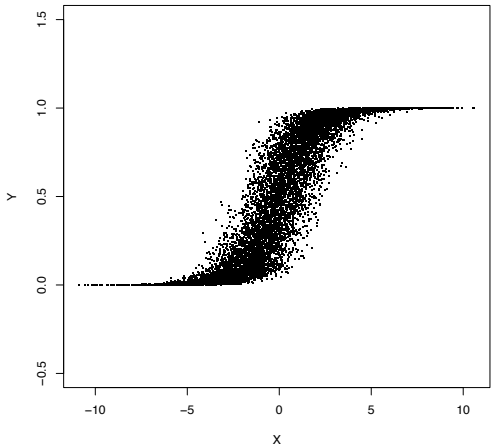
[1] http://github.com/xwshen51/engression

Numerical example

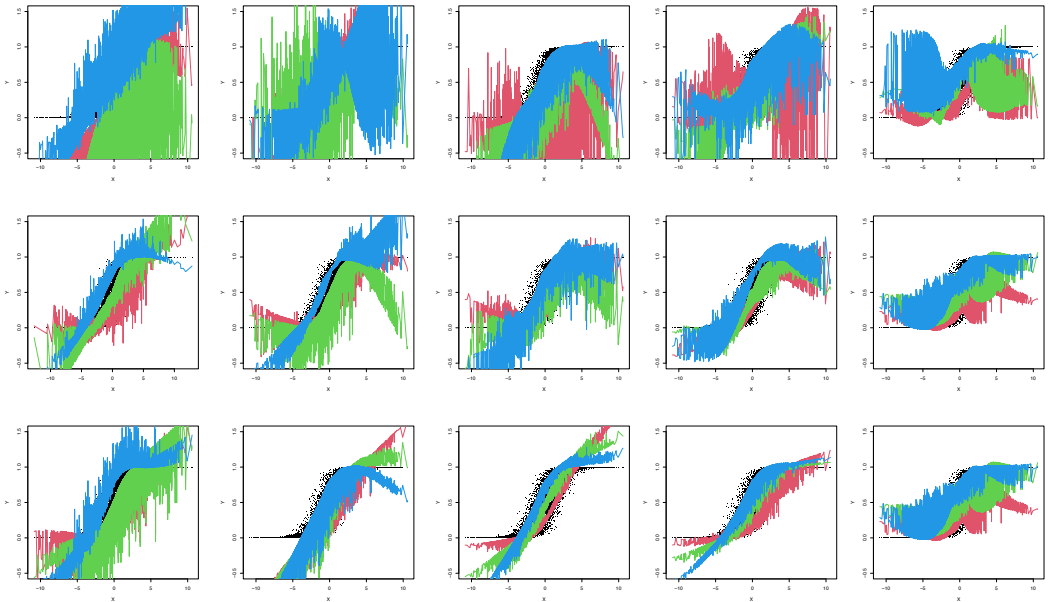
training data



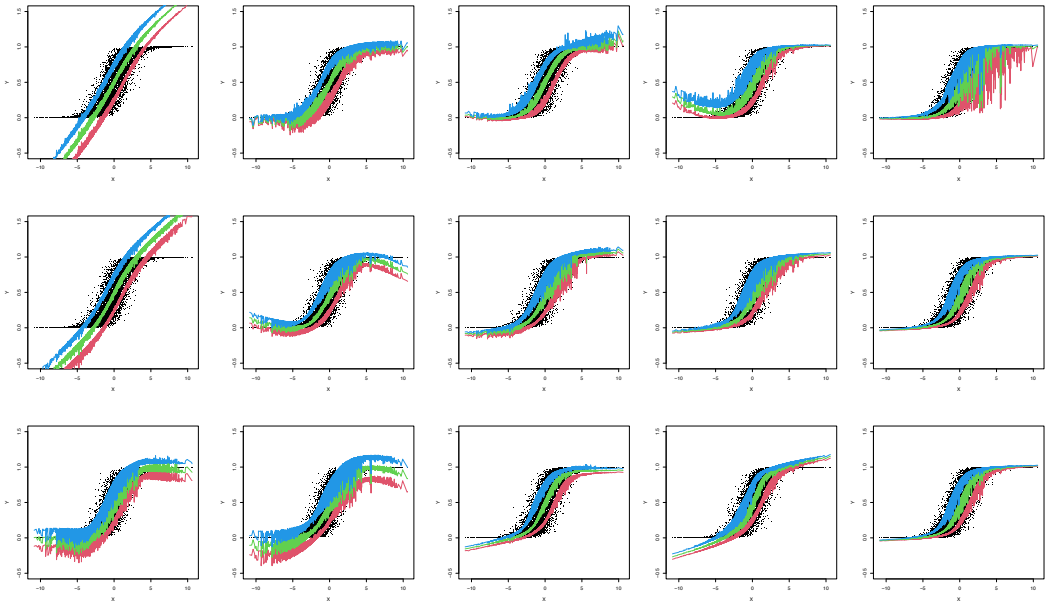
test data



NN quantile regression. Top to bottom: $10, 100$ and 1000 hidden dimension. Left to right: $2, 3, 5, 10$ and 20 layers.



Engression. Top to bottom: $10, 100$ and 1000 hidden dimension. Left to right: $2, 3, 5, 10$ and 20 layers.



Engression as a useful tool for climate prediction

- Ensemble-native predictions: directly generates large conditional ensembles rather than single deterministic outputs
- Full uncertainty quantification: captures variability, extremes, and dependence structures-not just averages
- Scalable to high-dimensional climate fields
- More expressive than classical statistical methods
- Simpler and computationally lighter than machine learning methods, robust to hyper-parameters

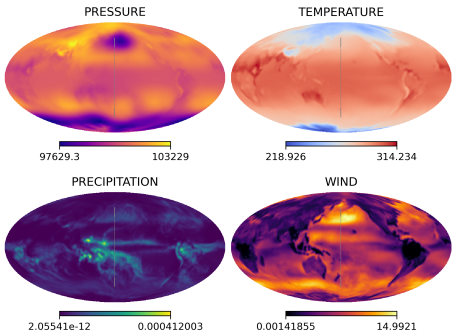
Part II Distributionally Lossless Dimension Reduction

Distributional Principal Autoencoders

with Nicolai Meinshausen

General circulation model emulation [1]

Response (about $500k$ variables): monthly pressure, temperature, precipitation, wind speed, resolved each on a 180×360 grid



Predictors (10-20 variables): month, stratospheric O3 radiative forcing, solar radiative forcing, volcanic radiative forcing, total O3 radiative forcing, aerosol radiative forcing, CO2 radiative forcing

[1] Joint with Nicolai Meinshausen and Malte Meinshausen

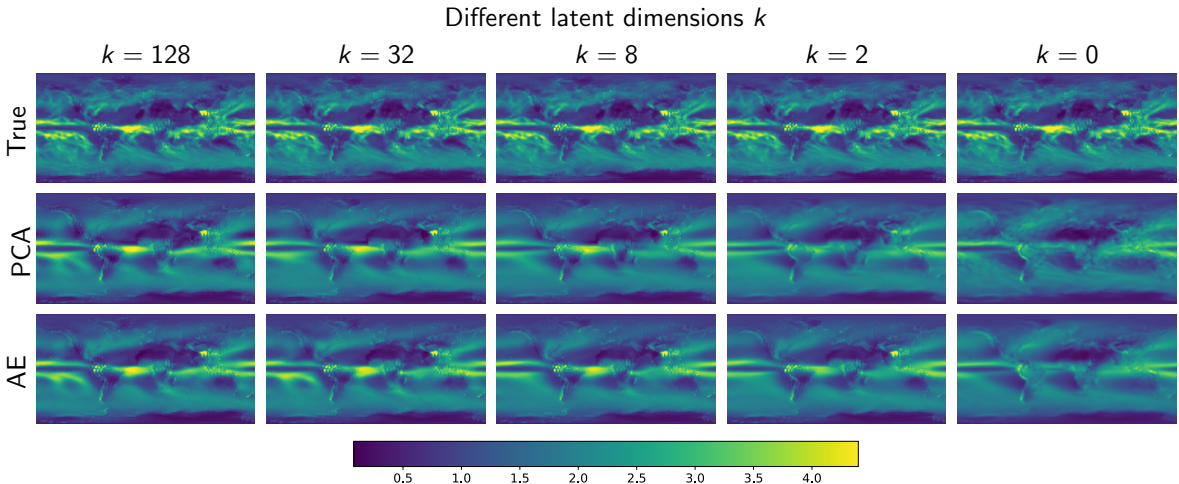
Dimension reduction

- Data $X \in \mathbb{R}^p$
- Dimension reduction: *encoder* $e(.) : \mathbb{R}^p \to \mathbb{R}^k$ where $k < p$.
- Data reconstruction: *decoder* $d(.) : \mathbb{R}^k \to \mathbb{R}^p$.
- Common criterion: minimising the mean squared reconstruction loss

$$\min \mathbb{E}\left[\|X - d(e(X))\|^2\right]$$

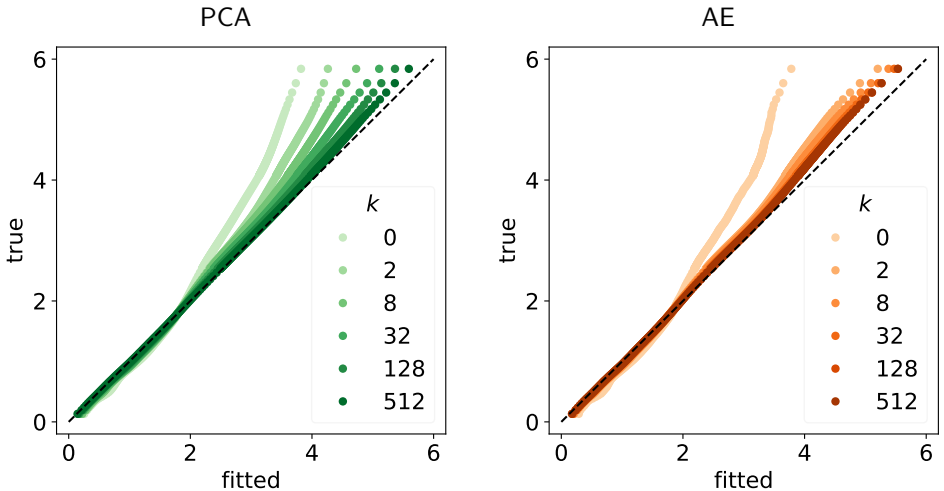
- Examples:
  - Principal Component Analysis (PCA): linear encoder and decoder
  - Autoencoders (AE): neural network encoder and decoder
- Lossy compression: when $k < p$, we typically have $X \neq d(e(X))$.

PCA and AE reconstructions smooth the climate fields—loss of spatial variability and extremes



Reconstructions for global monthly precipitation fields (square-root transformed, original unit $\mathrm{kg} \cdot \mathrm{m}^{-2}\mathrm{s}^{-1}$)

Q–Q plots of precipitations at a random location for test data versus fitted distributions



PCA and AE reconstructions under-estimate the high quantiles.

Distributional reconstruction

- Mean reconstruction (autoencoders):

$$d(z) = \mathbb{E}[X|e(X) = z], \ \forall z.$$

- Distributional reconstruction (ours):

$$d(z, \varepsilon) \stackrel{d}{=} \big(X|e(X) = z\big), \ \forall z.$$

$\Rightarrow$ Distributionally lossless compression:

$$d(e(X), \varepsilon) \stackrel{d}{=} X$$

irrespective of the latent dimension.

Distributionally lossless compression

Statistical meaning: reconstructions share the same distribution as the original data

$$d(e(X), \varepsilon) \overset{d}{=} X$$

Physical meaning:

- No artificial variance damping or inflation
- Extremes are preserved even under aggressive compression
- Downstream statistics match the original climate model

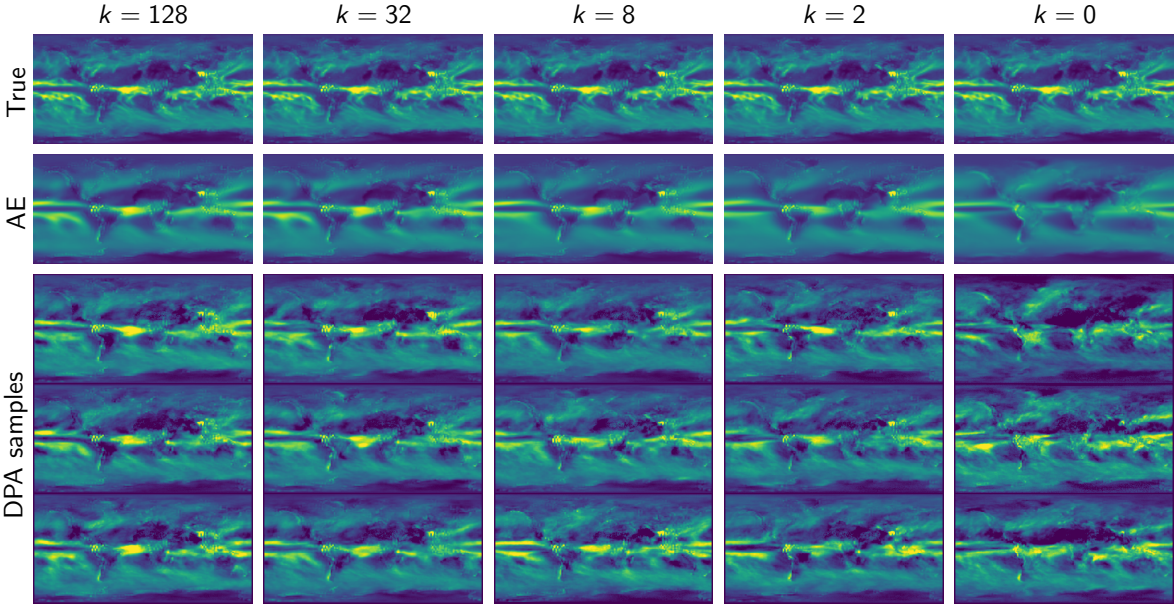
Distributional Principal Autoencoder (DPA)

To achieve distributional reconstruction, i.e.

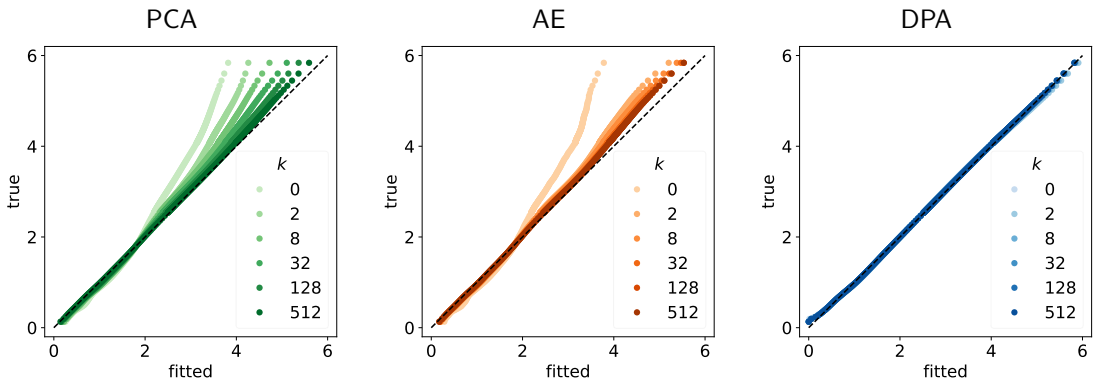
$$d(z, \varepsilon) \overset{d}{=} \big(X | e(X) = z\big), \ \forall z.$$

DPA solution (engression applied to $X|e(X)$):

$$\underset{e,d}{\operatorname{argmin}} \, \mathbb{E}\Big[\|X - d(e(X), \varepsilon)\| - \frac{1}{2}\|d(e(X), \varepsilon) - d(e(X), \varepsilon')\|\Big]$$


$k = 128$
$k = 32$
$k = 8$
$k = 2$
$k = 0$
True
AE
DPA samples

Q–Q plots of precipitations at a random location for test data versus fitted distributions



DPA reconstructions get the high quantiles right.

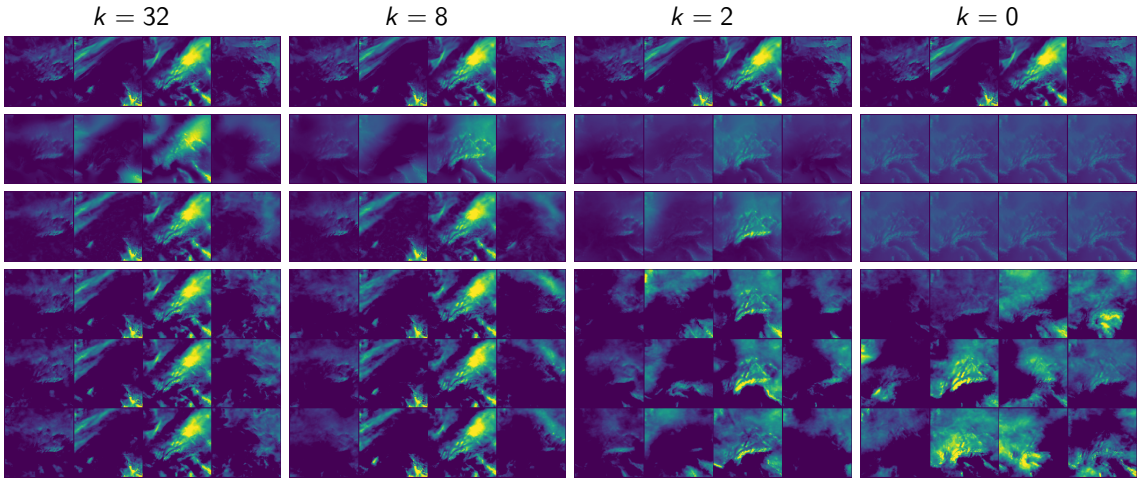
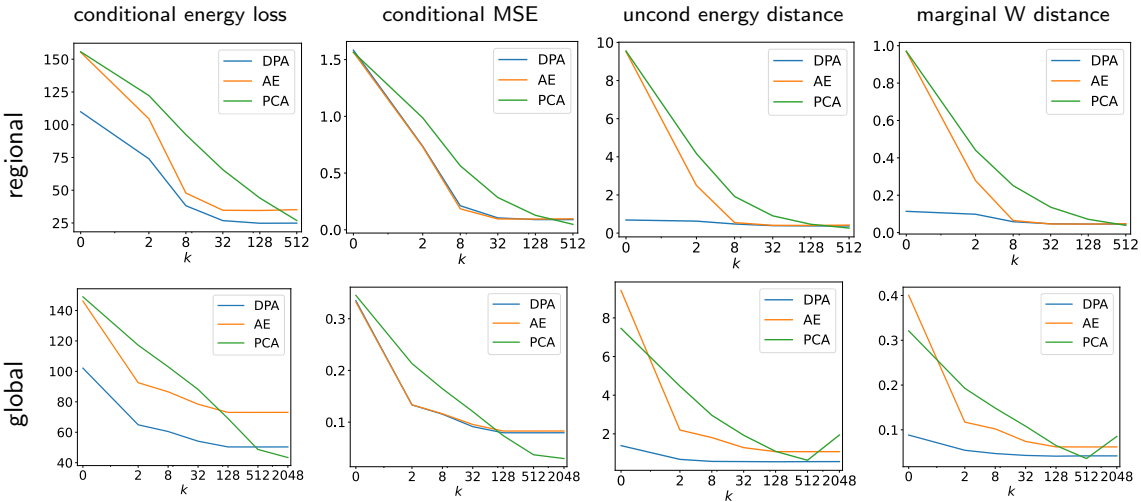
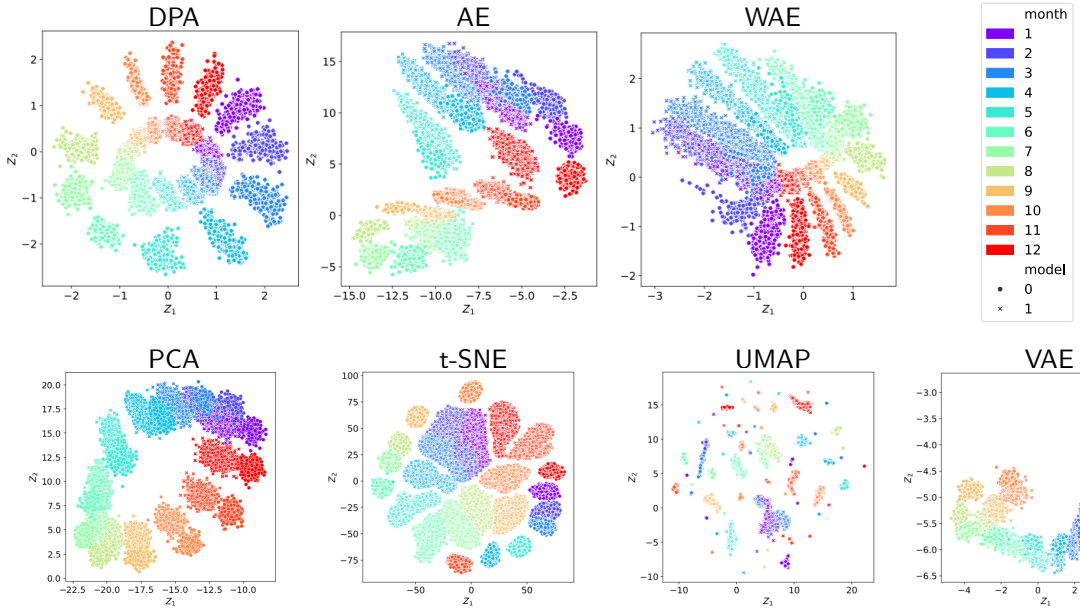


Figure: Regional monthly precipitation in central Europe with a dimension of 128×128.

Regional and global precipitations



2D visualization for global precipitation fields (original dimension: 360×180)



Methodological ingredients

Challenges in climate emulation:

- Distributional $\Leftarrow$ engression for learning distributions
- High-dimensional $\Leftarrow$ DPA for distribution-preserved dimension reduction

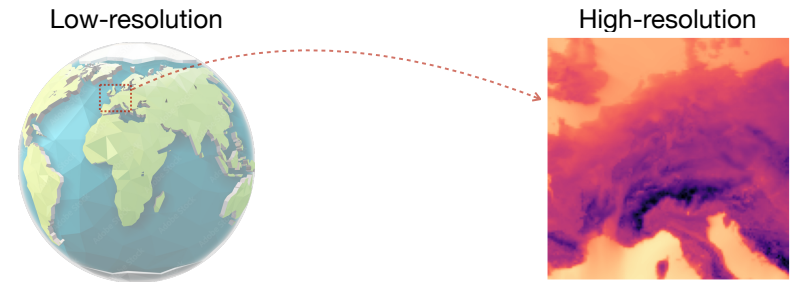
Part III Statistical Downscaling

EnScale: Temporally-consistent multivariate generative downscaling via proper scoring rules

with Maybritt Schillinger, Maxim Samarin,
Reto Knutti, and Nicolai Meinshausen

Statistical Downscaling

Goal: emulating RCM

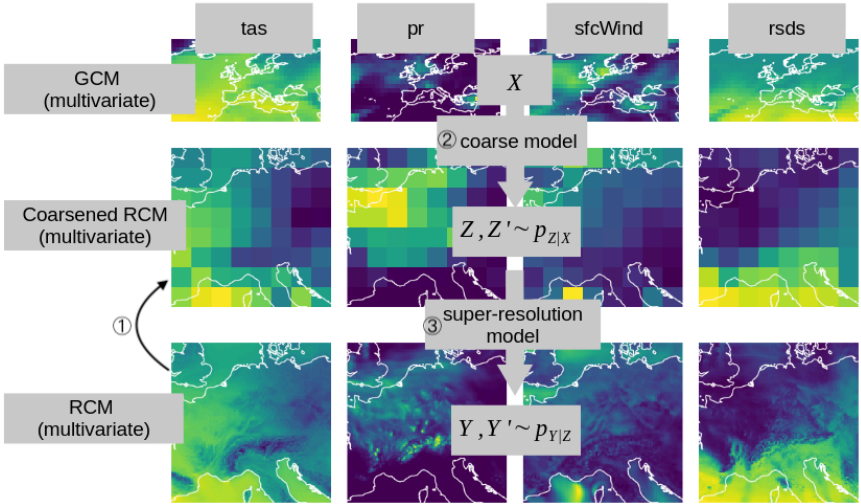


Statistical goal: estimating $P_{Y|X}$

Three-step approach

1. Reduce the dimension of regional high-resolution fields Y to lower-dimensional one Z by *spatial coarsening (fixed encoder)*
2. Learn *engression* to predict Z from low-resolution fields X
3. Reconstruct high-resolution fields Y back from predicted Z by *DPA*

EnScale architecture



- DPA reconstruction for super-resolution from coarsened RCM $Z = e^*(Y)$ to Y

$$d^* \in \operatorname*{argmin}_d \mathbb{E}\Big[\|Y - d(e^*(Y), \eta)\| - \frac{1}{2}\|d(e^*(X), \eta) - d(e^*(X), \eta')\|\Big]$$

  We have $d^*(z, \eta) \sim P_{Y|e^*(Y)=z}$.
- Engression for the coarsened RCM Z on GCM X

$$g^* \in \operatorname*{argmin}_g \mathbb{E}\Big[\|Z - g(X, \varepsilon)\| - \frac{1}{2}\|g(X, \varepsilon) - g(X, \varepsilon')\|\Big]$$

  We have $g^*(x, \varepsilon) \sim P_{Z|X=x}$.
- Final prediction given an RCM data X:

$$d^*(g^*(X, \varepsilon), \eta) \sim P_{Y|X=x} \text{ approximately}$$

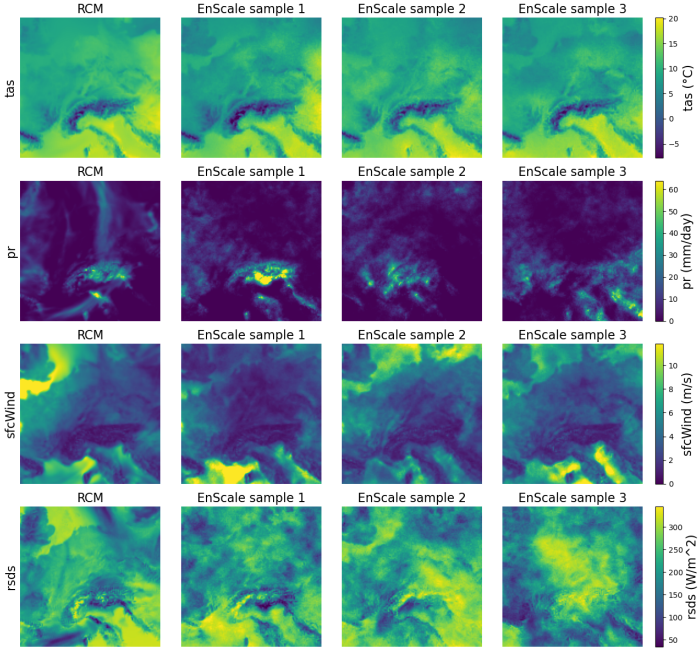
Data

- 8 RCM-GCM pairs from EURO-CORDEX

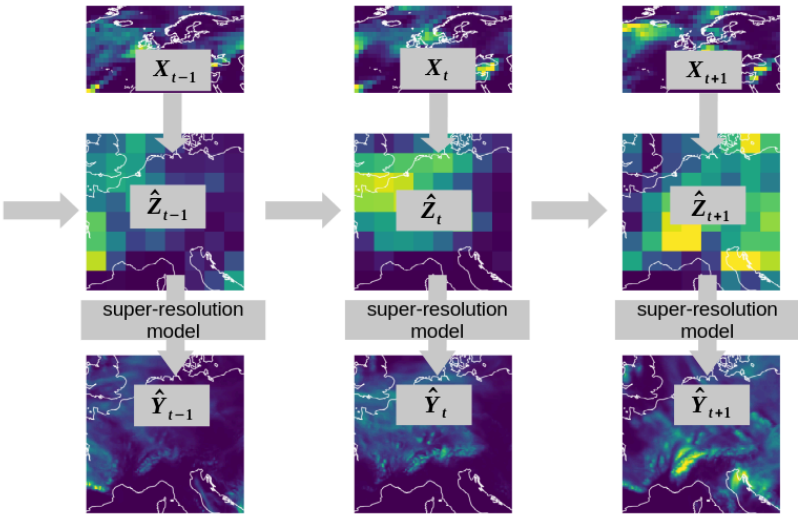
RCM	**GCM**
ALADIN63	CNRM-CM5
ALADIN63	MPI-ESM-LR
CCLM4-8-17	CNRM-CM5
CCLM4-8-17	MIROC5
CCLM4-8-17	MPI-ESM-LR
REMO2015	MIROC5
RegCM4-6	CNRM-CM5
RegCM4-6	MPI-ESM-LR

- Variables: daily 2m temperature (*tas*), total precipitation (*pr*), surface wind (*sfcWind*) and surface downwelling shortwave (solar) radiation (*rsds*)
- Additional predictor: sea level pressure

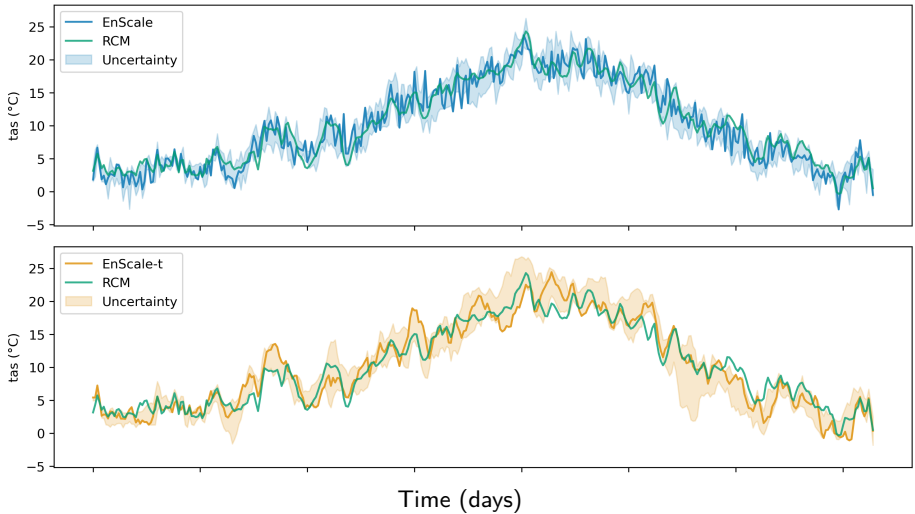
EnScale samples



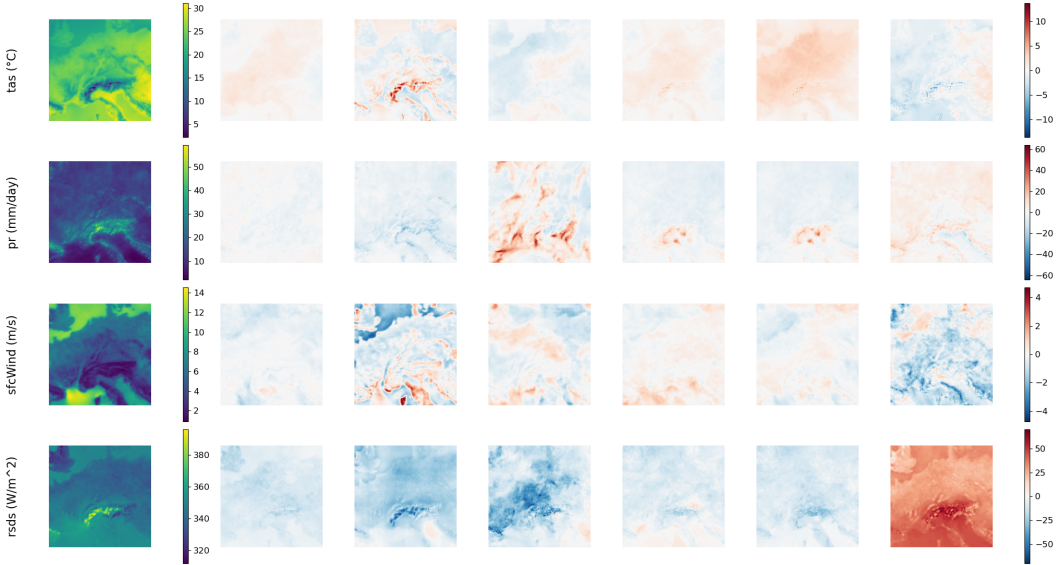
EnScale-t for temporal consistency



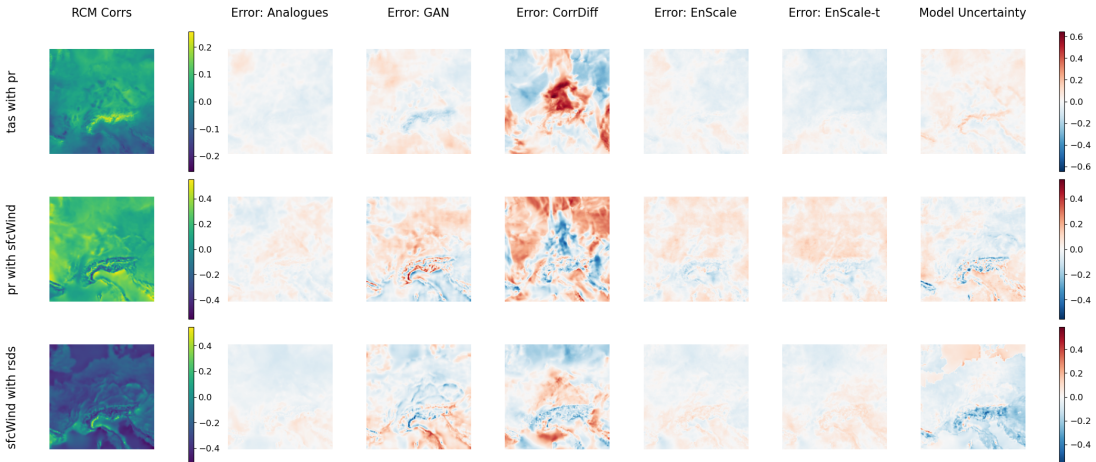
Temporal consistency



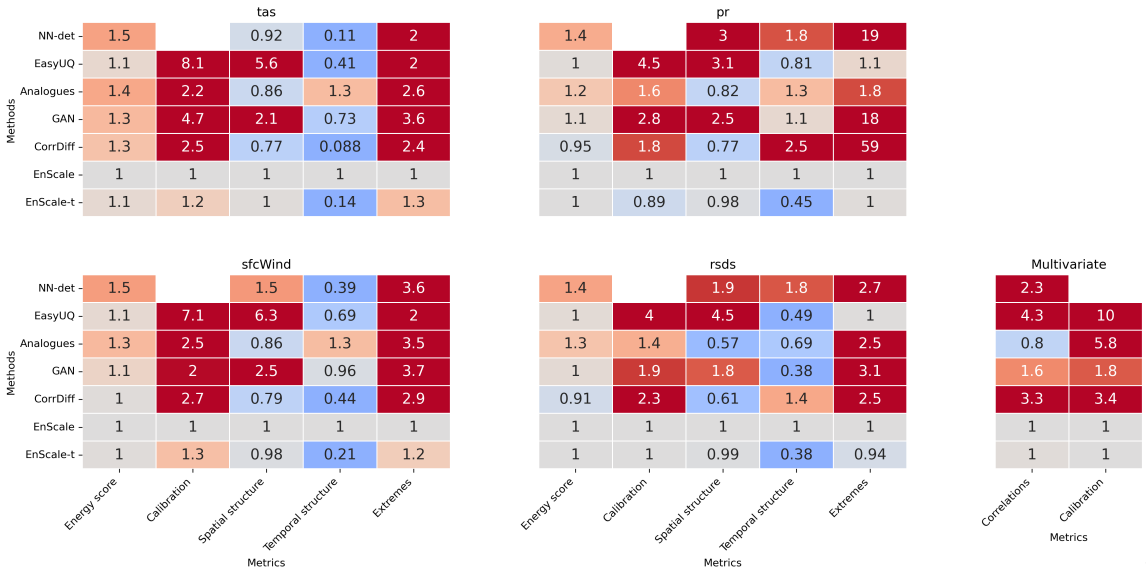
Extremes 95% (temperal) quantiles



Correlations between pairs of variables



Overall metrics



Computational cost

Method	Training	Inference
EnScale	24 h	1.75 h
CorrDiff	216 h	36 h
RCM	weeks	

Table: All runtimes were checked on a single NVIDIA A100 GPU (80GB) and exclude data loading. Inference times refer to producing a high-resolution ensemble of 100 years for a single GCM-RCM pair with 10 samples per day. RCMs are run on high-end CPUs.

Summary

Challenges in climate emulation: distributional and high-dimensional

Our methodological contributions:

- *Engression* for learning distributions[1]
- *DPA* for dimension reduction with distribution-preserving reconstruction[2]

Final product:

- *EnScale* for statistical downscaling[3]

Thank you!

[1] S. and Meinshausen, "Engression: Extrapolation through the Lens of Distributional Regression," *JRSSB*, 2024

[2] S. and Meinshausen, "Distributional Principal Autoencoders," arXiv:2404.13649

[3] Schillinger, Samarin, S., Knutti, Meinshausen, "EnScale: Temporally-consistent multivariate generative downscaling via proper scoring rules," arXiv:2509.26258