AI for High-Resolution Regional Climate Modeling

Advanced School on
High-Performance Computing and
Applied AI for High-Resolution Regional
Climate Modeling

Jose González-Abad
Postdoctoral Researcher
gonzabad@ifca.unican.es

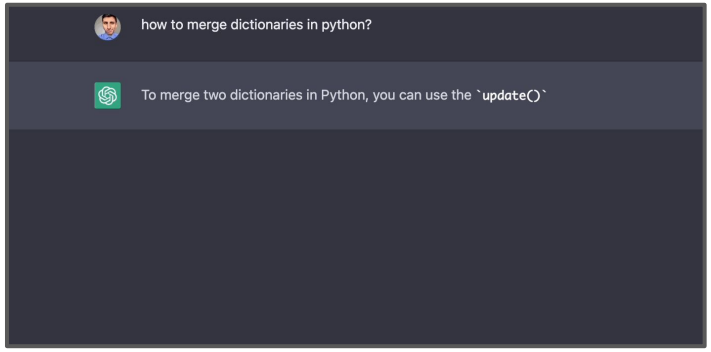
Table of Contents

Table of Contents

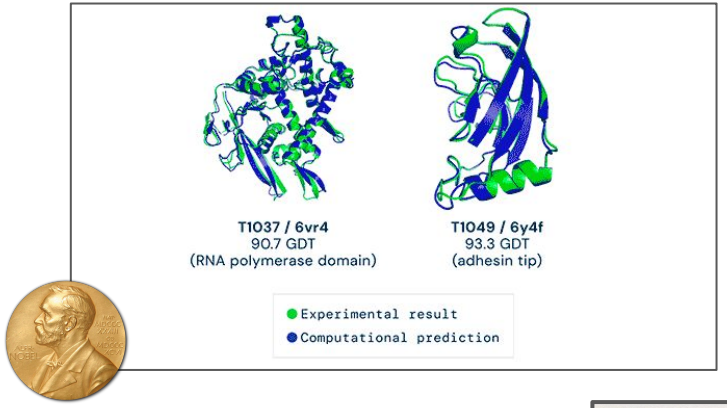
Deep Learning Revolution

Deep learning models have **revolutionized many fields**, achieving results that were never imagined a few years ago.

LLMs

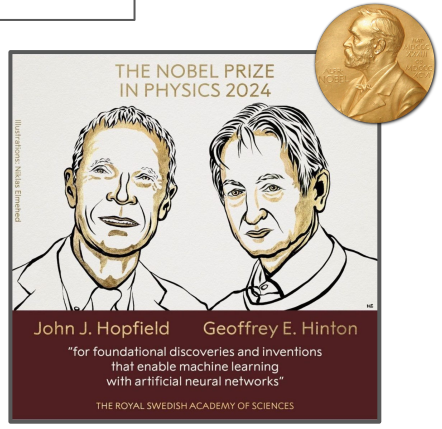


Protein Folding



Video Generation

Mastering of Go



Deep Learning Revolution

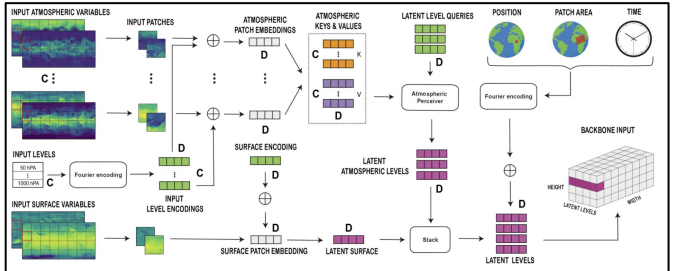


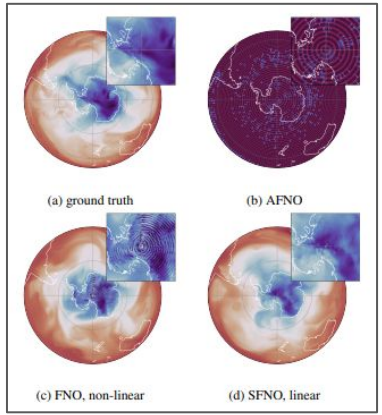
Figure 2: An illustration of Aurora's encoder module. Input weather states are tokenized and compressed into a 3D latent representation using Perceiver-style cross-attention blocks. The resulting latent tokens are augmented with appropriate encodings that provide spatial, temporal, and scale information.

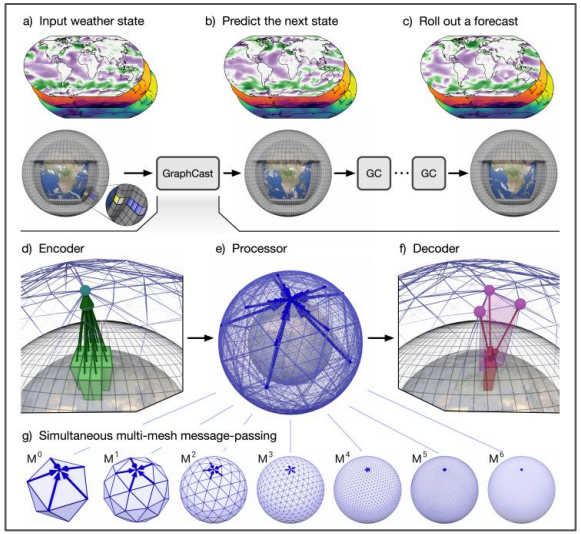
A Foundation Model for the Earth System

Cristian Bodnar[1,2†], Wessel P. Bruinsma[1†], Ana Lucic[1,3†], Megan Stanley[1†], Anna Vaughan[4†], Johannes Brandstetter[1,5], Patrick Garvan[1], Maik Riechert[1], Jonathan A. Weyn[6], Haiyu Dong[6], Jayesh K. Gupta[2,7], Kit Thambiratnam[6], Alexander T. Archibald[4], Chun-Chieh Wu[8], Elizabeth Heider[1], Max Welling[1,3], Richard E. Turner[1,4,9], Paris Perdikaris[1,10*]

Spherical Fourier Neural Operators: Learning Stable Dynamics on the Sphere

Boris Bonev [1] Thorsten Kurth [1] Christian Hundt [1] Jaideep Pathak [1] Maximilian Baust [1] Karthik Kashinath [1] Anima Anandkumar [1 2]

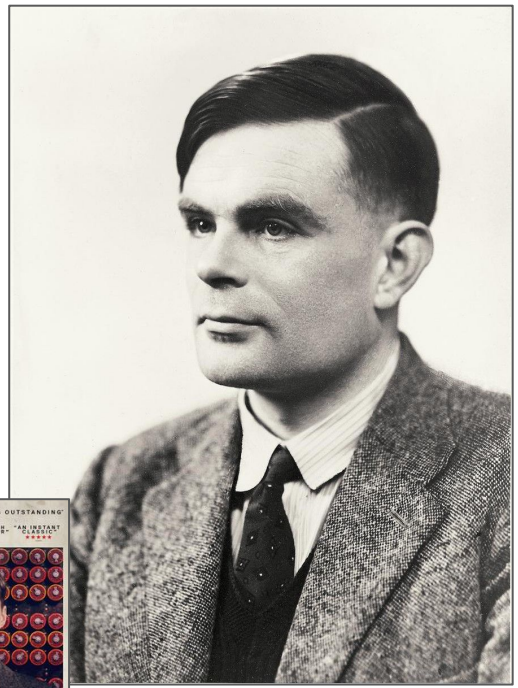




GraphCast: Learning skillful medium-range global weather forecasting

Remi Lam[*,1], Alvaro Sanchez-Gonzalez[*,1], Matthew Willson[*,1], Peter Wirnsberger[*,1], Meire Fortunato[*,1], Ferran Alet[*,1], Suman Ravuri[*,1], Timo Ewalds[1], Zach Eaton-Rosen[1], Weihua Hu[1], Alexander Merose[2], Stephan Hoyer[2], George Holland[1], Oriol Vinyals[1], Jacklynn Stott[1], Alexander Pritzel[1], Shakir Mohamed[1] and Peter Battaglia[1]
[*]equal contribution, [1]Google DeepMind, [2]Google Research

Weather Turing Test



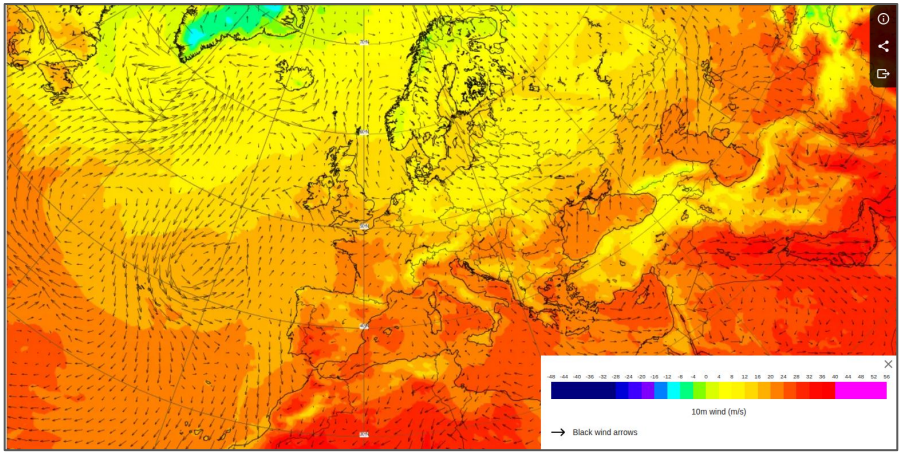
Alan Turing

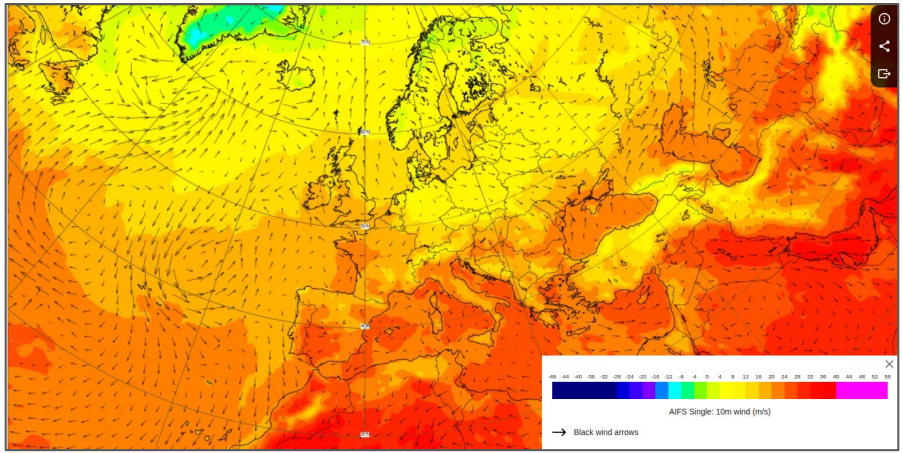
The **Turing Test**, proposed by Alan Turing in 1950, is a benchmark for **determining whether a machine can exhibit intelligent behavior indistinguishable from that of a human**. In the test, a human evaluator interacts with both a machine and a human without knowing which is which. **If the evaluator cannot reliably tell them apart, the machine is said to have passed the test.**

Weather Turing Test

Which one comes from a **physics-based** model and which from a **DL-based** one?

2m temperature and 10m wind

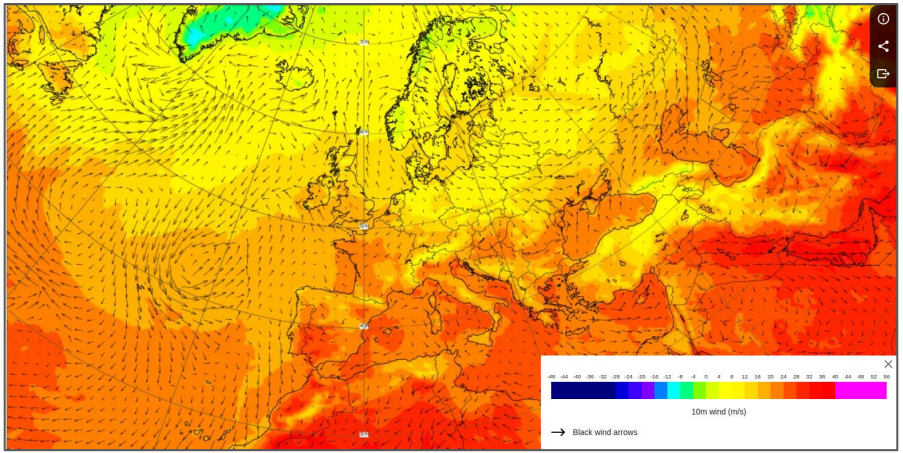




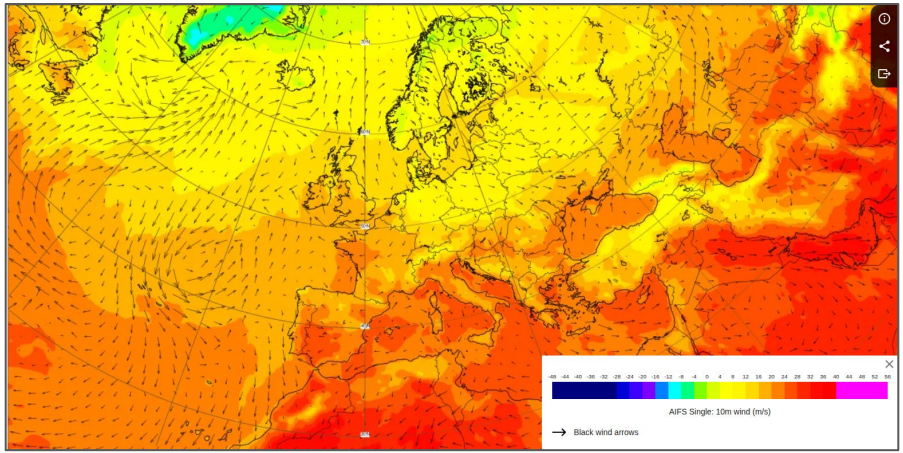
Weather Turing Test

Which one comes from a **physics-based** model and which from a **DL-based** one?

2m temperature and 10m wind



Physics-based model

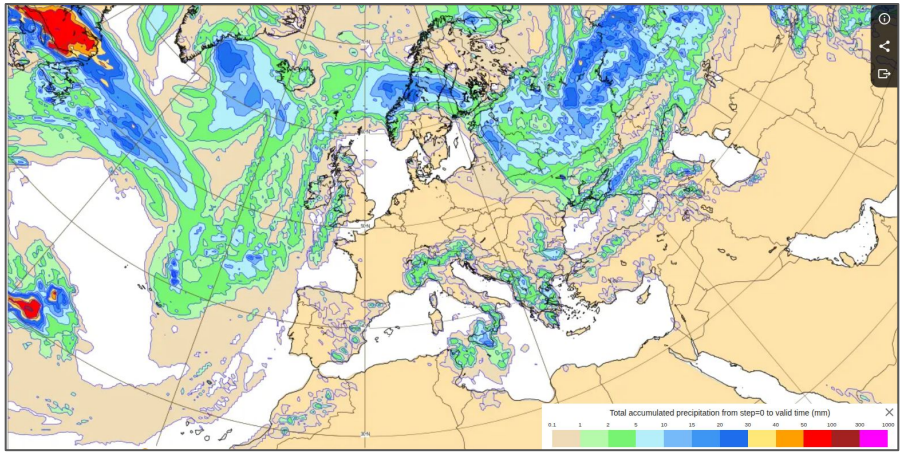


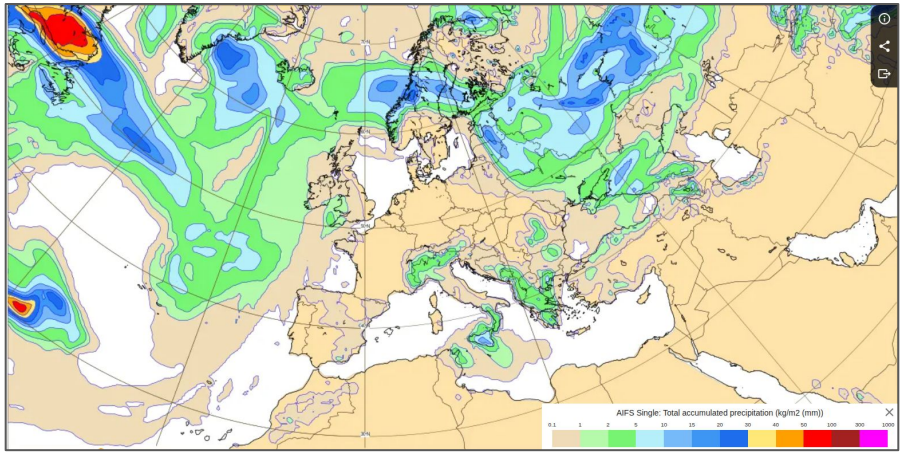
AIFS (DL-based model)

Weather Turing Test

Which one comes from a **physics-based** model and which from a **DL-based** one?

Accumulated Precipitation

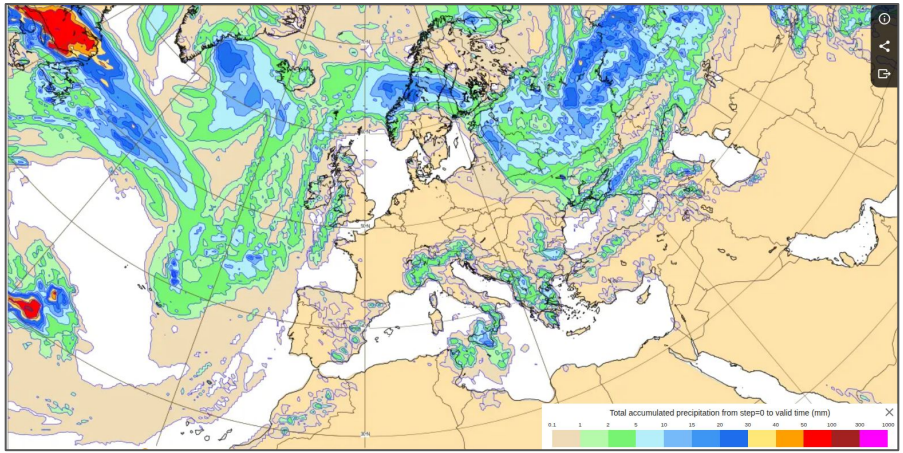




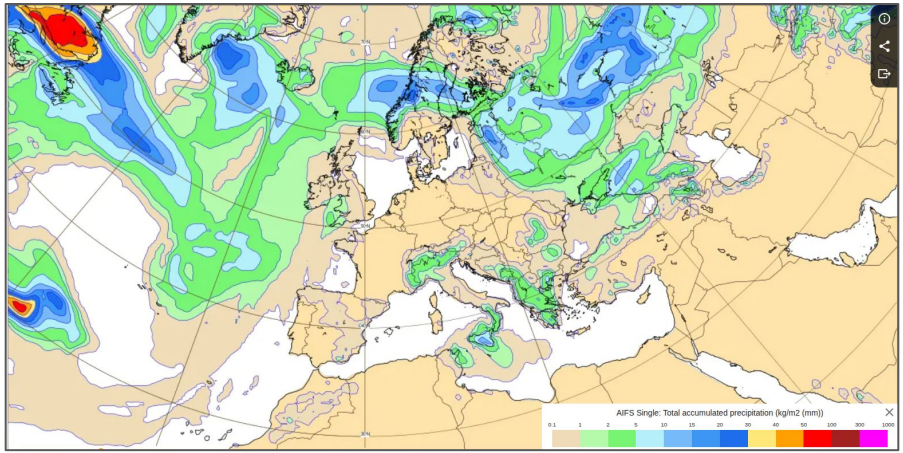
Weather Turing Test

Which one comes from a **physics-based** model and which from a **DL-based** one?

Accumulated Precipitation



Physics-based model

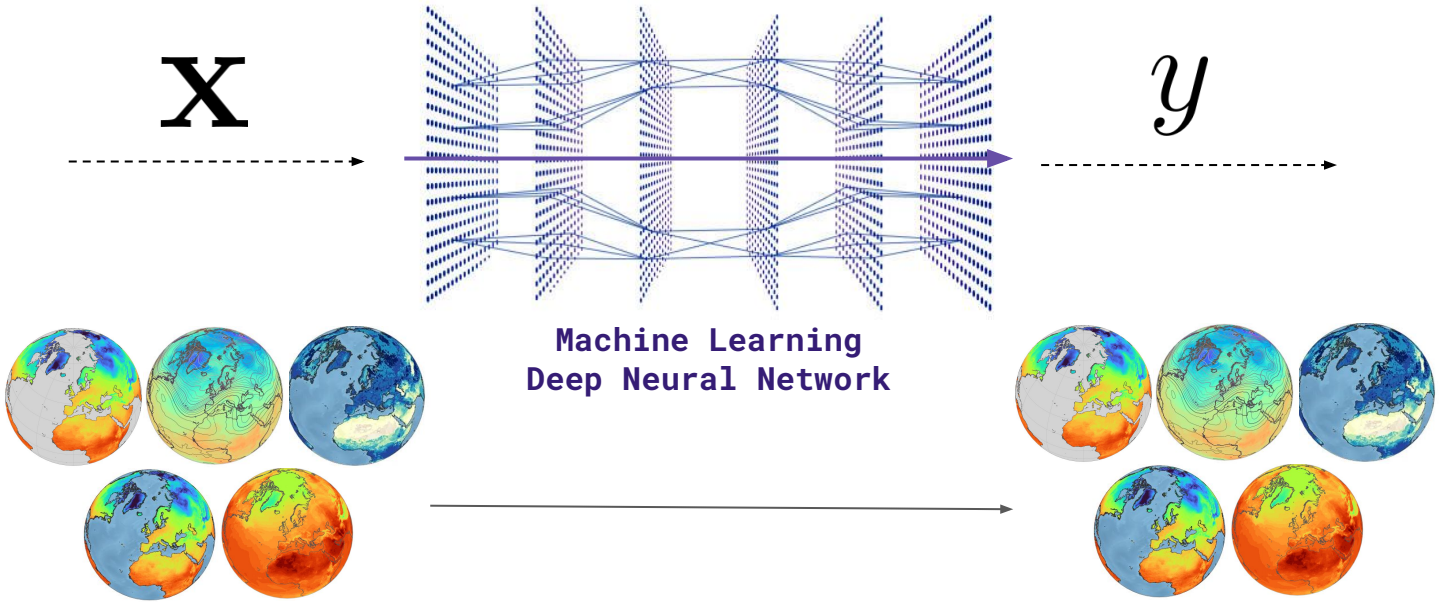


AIFS (DL-based model)

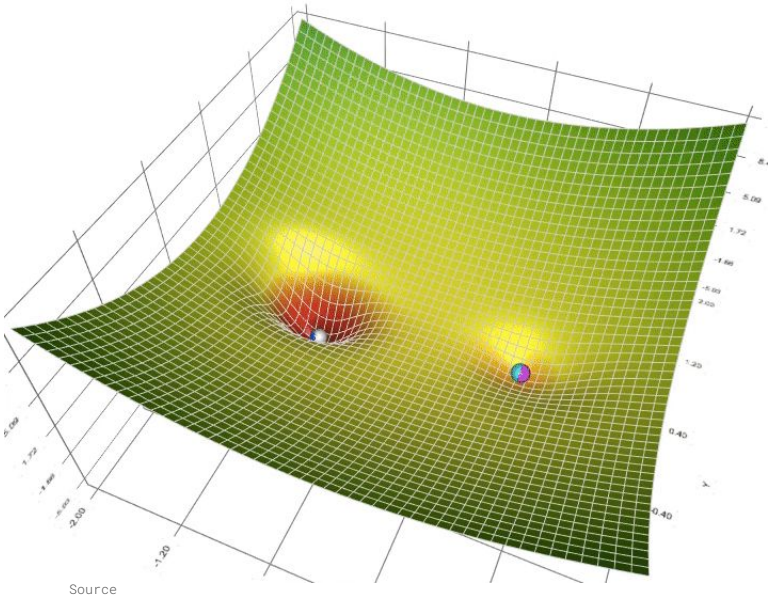
Weather Turing Test

DL-based models tend to generate **smoothed/blurry predictions**

Deep Learning



Deep Learning



Source

We adjust the parameters of the neural network to minimize the value of a loss function associated with the task we want to accomplish

The most typical is the **Mean Squared Error** (MSE)

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i\right)^2$$

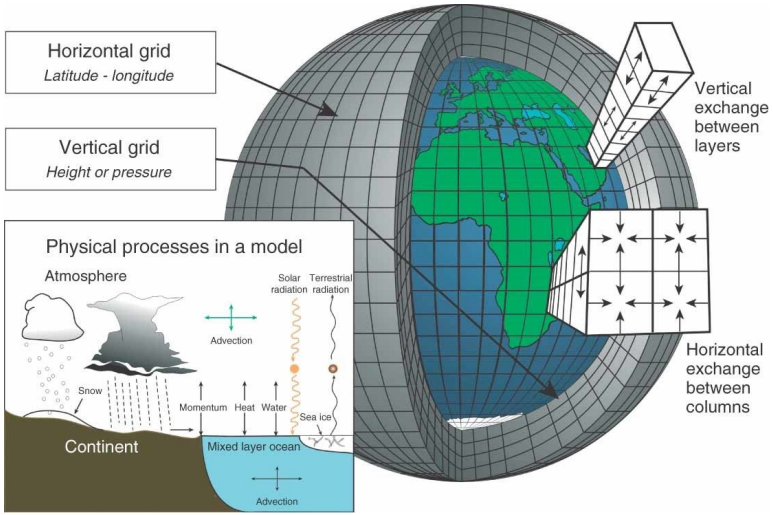
Global Climate Models

Global Climate Models (GCM) are the main tools used to **simulate the evolution of climate** by numerically solving the physical equations characterizing climate dynamics:

Climate equations

$$
\begin{aligned}
\frac{d\boldsymbol{v}}{dt} &= -\alpha\boldsymbol{\nabla}p - \boldsymbol{\nabla}\phi + \boldsymbol{F} - 2\boldsymbol{\Omega}\times\boldsymbol{v} \\
\frac{\partial\rho}{\partial t} &= -\boldsymbol{\nabla}\cdot(\rho\,\boldsymbol{v}) \\
p\,\alpha &= RT \\
Q &= C_p\frac{dT}{dt} - \alpha\frac{dp}{dt} \\
\frac{\partial\rho\,q}{\partial t} &= -\boldsymbol{\nabla}\cdot(\rho\,\boldsymbol{v}\,q) + \rho(E-C)
\end{aligned}
$$

discretize over space and time →



Source: Edwards, P. N. (2011). History of climate modeling. Wiley Interdisciplinary Reviews: Climate Change, 2(1), 128-139.

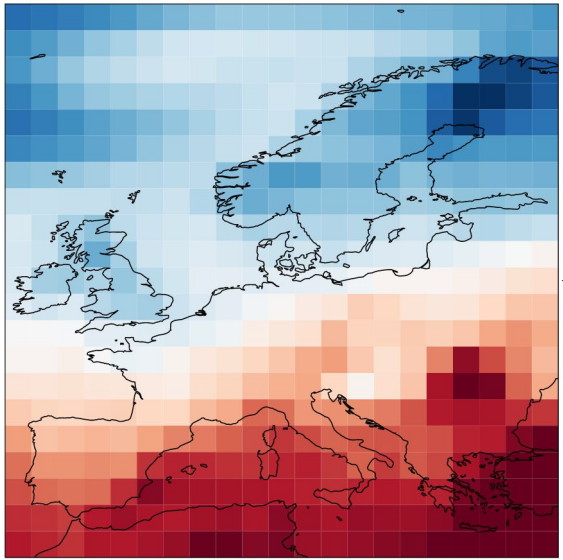
With **typical resolutions** of hundreds of km (e.g. **~100 km** in CMIP6)

Coarse Resolution of GCMs

Due to deficiencies in the climate modeling process, **GCMs can not accurately reproduce the local scale**

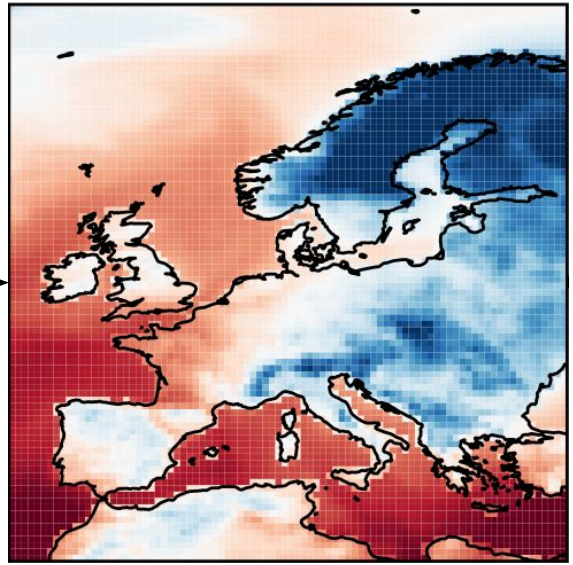
This makes it difficult to use GCMs in different socio-economical activities where **regional information** is required

GCM output



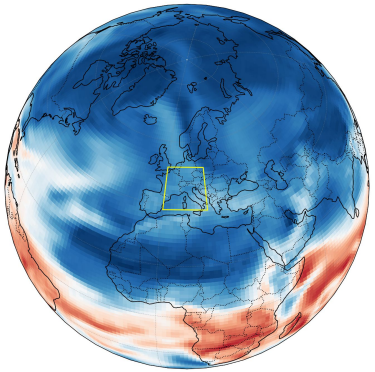
How can we avoid this **difference in resolution**?

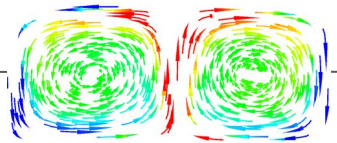
Observations



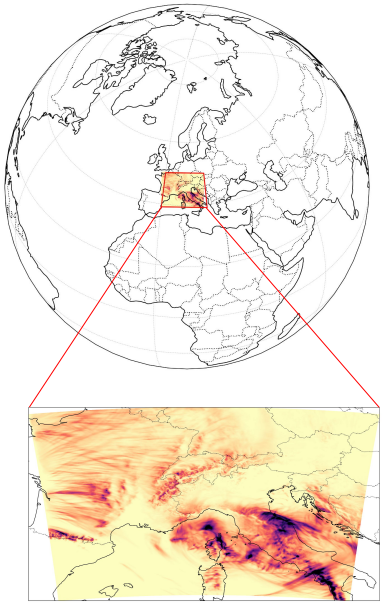
Regional Climate Models

Global Climate Model (GCM)
CNRM-CM5 (**~100 km**)



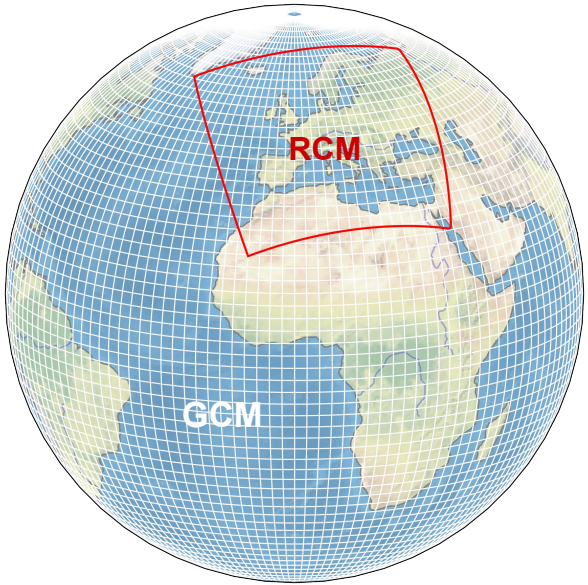


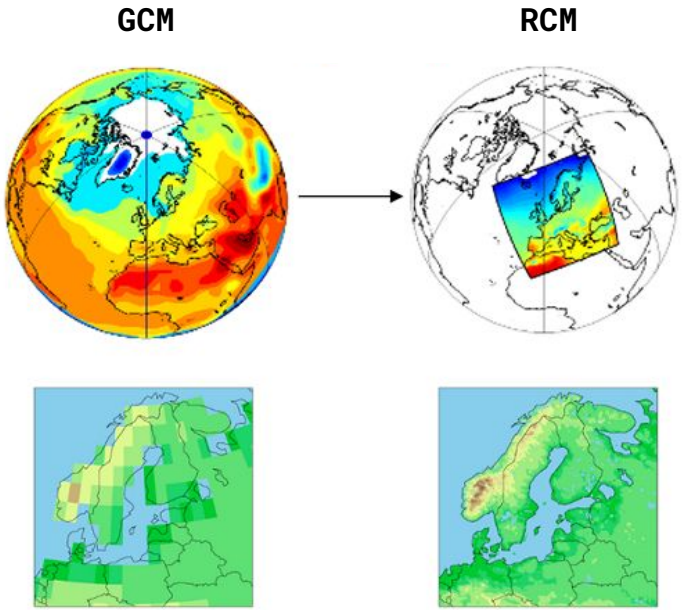
Regional Climate Model (RCM)
AROME (**~3 km**)



A RCM is a **high-resolution physics-based model focused on a specific region**, using boundary conditions from a GCM.

Regional Climate Models



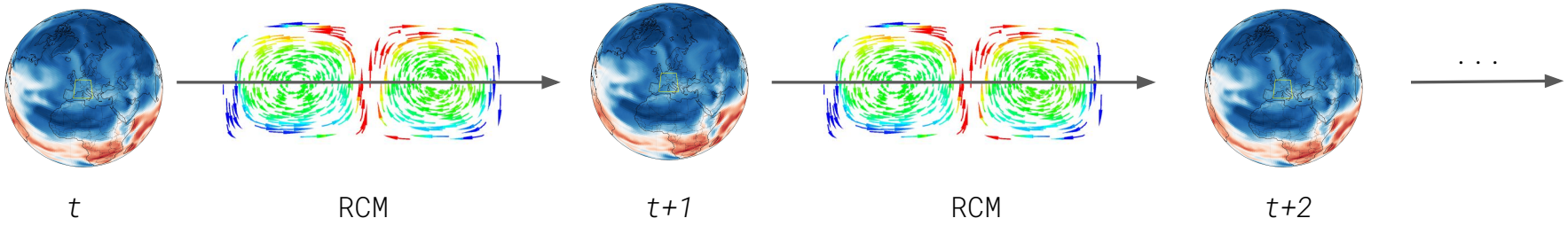


Source

A RCM is a **high-resolution physics-based model focused on a specific region**, using boundary conditions from a GCM.

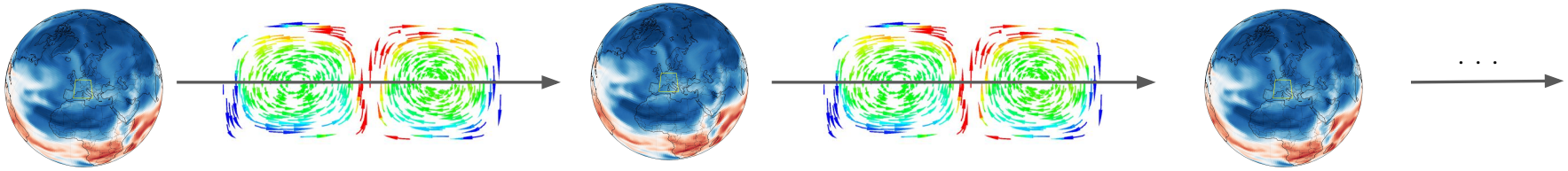
Regional Climate Models

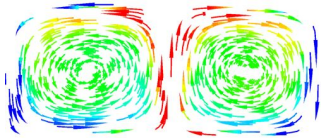
Running a RCM is **computationally expensive** and **challenging**



Regional Climate Models

Running a RCM is **computationally expensive** and **challenging**

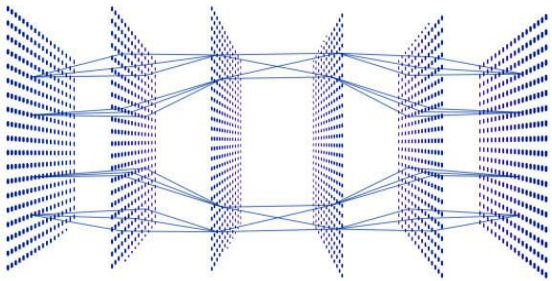




requires

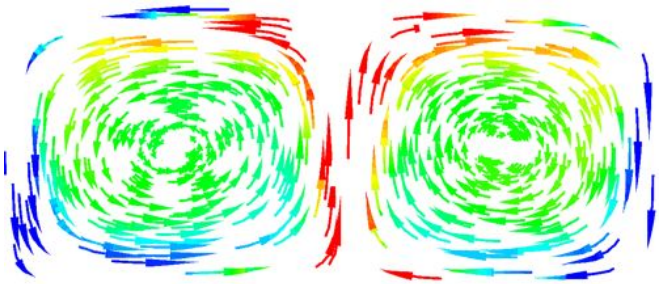
RCM Emulation

What if we **emulate** the RCM with a Deep Learning model?



Deep Learning

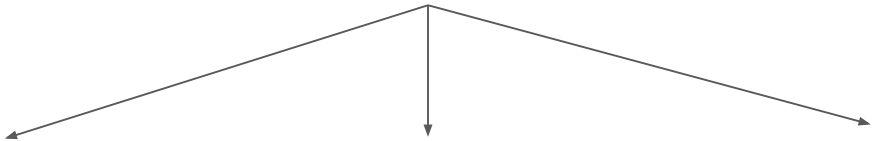
$\sim$

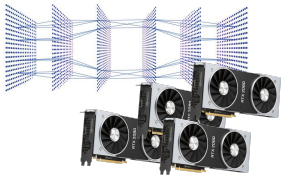


RCM

Benefits of RCM Emulation

The main point of the emulation of RCMs is to better **assess the uncertainty linked to future projections**

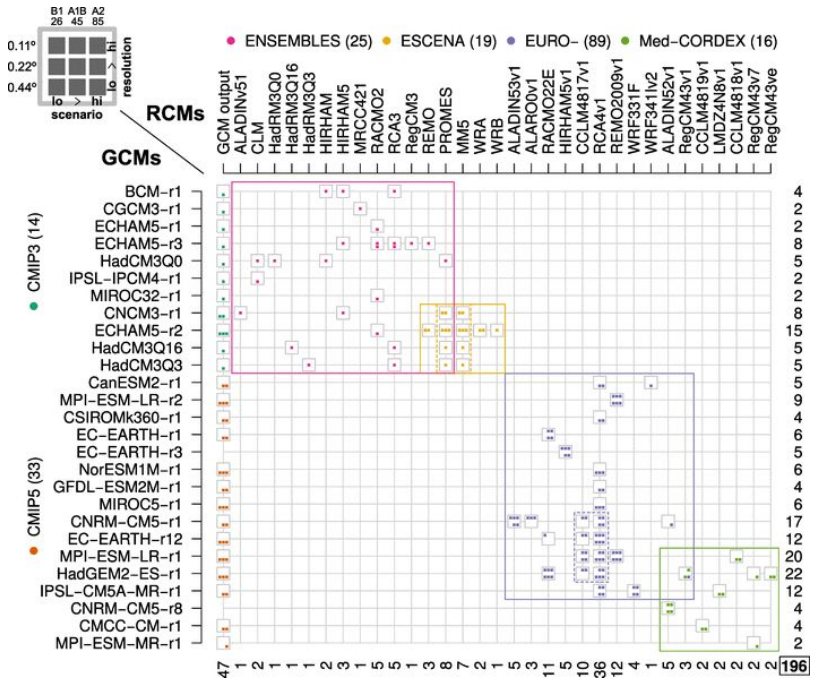




vs

Benefits of RCM Emulation

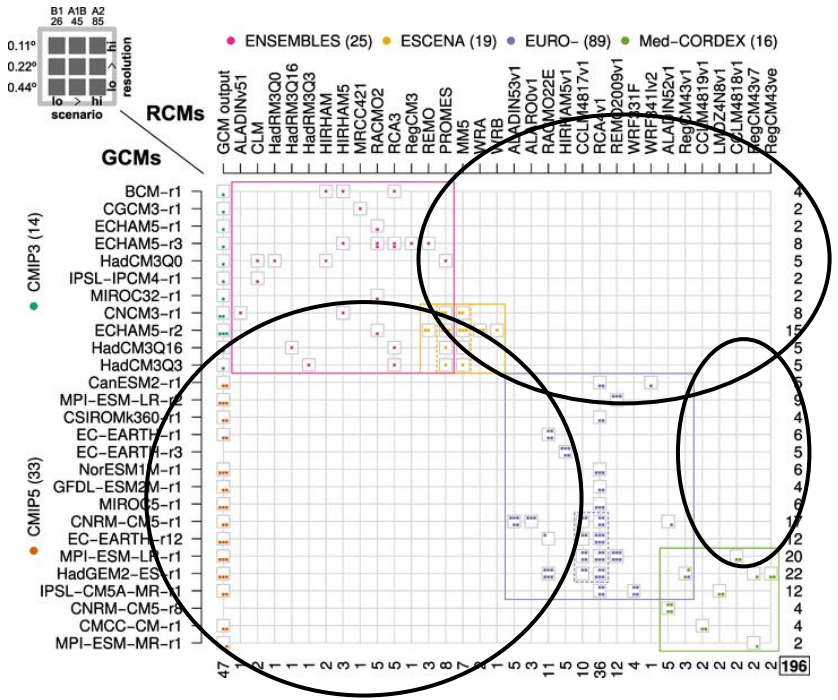
These emulators can help us **fill the GCM/RCM matrix**



Fuente: Fernández, J., Frías, M. D., Cabos, W. D., Cofiño, A. S., Domínguez, M., Fita, L., ... & Sánchez, E. (2019). Consistency of climate change projections from multiple global and regional model intercomparison projects. Climate dynamics, 52, 1139-1156.

Benefits of RCM Emulation

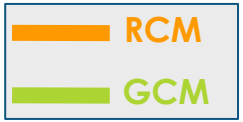
These emulators can help us **fill the GCM/RCM matrix**



Fuente: Fernández, J., Frías, M. D., Cabos, W. D., Cofiño, A. S., Domínguez, M., Fita, L., ... & Sánchez, E. (2019). Consistency of climate change projections from multiple global and regional model intercomparison projects. Climate dynamics, 52, 1139-1156.

How are RCM emulators trained?

How can we emulate a RCM with DL? *

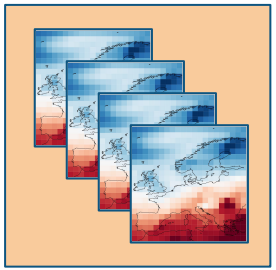


* RCM emulation generally focuses on local-scale variables (e.g., temperature and precipitation)

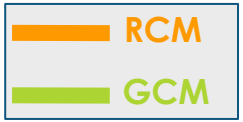
How are RCM emulators trained?

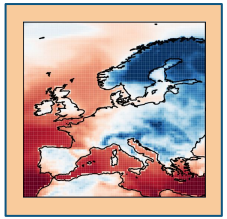
How can we emulate a RCM with DL? *

Large-scale variables
(Low-resolution)



Air-temperature
Specific Humidity
Wind Components
Geopotential





Temperature
Precipitation

Surface variable
(High-resolution)

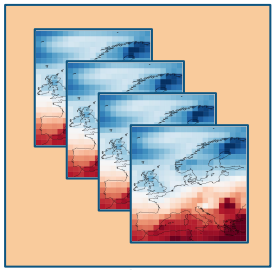
* RCM emulation generally focuses on local-scale variables (e.g., temperature and precipitation)

How are RCM emulators trained?

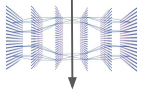
How can we emulate a RCM with DL? *

Large-scale variables
(Low-resolution)

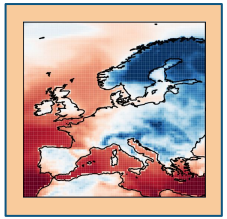
$\mathbf{x}$



Air-temperature
Specific Humidity
Wind Components
Geopotential

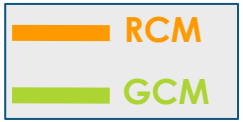


y



Temperature
Precipitation

Surface variable
(High-resolution)



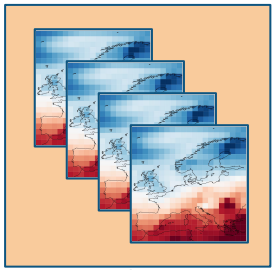
* RCM emulation generally focuses on local-scale variables (e.g., temperature and precipitation)

How are RCM emulators trained?

How can we emulate a RCM with DL? *

Large-scale variables
(Low-resolution)

$\mathbf{x}$

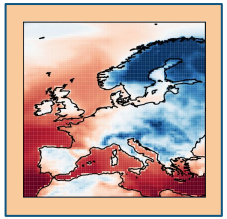


Air-temperature
Specific Humidity
Wind Components
Geopotential

RCM
GCM

Downscaling Function

y



Temperature
Precipitation

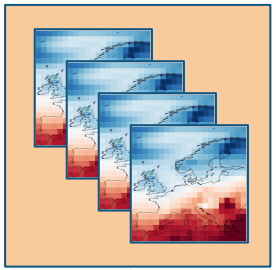
Surface variable
(High-resolution)

* RCM emulation generally focuses on local-scale variables (e.g., temperature and precipitation)

How are RCM emulators trained?

How can we emulate a RCM with DL? *

Large-scale variables
(Low-resolution)

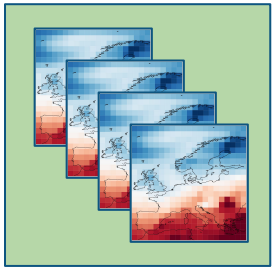


$\mathbf{x}$

Air-temperature
Specific Humidity
Wind Components
Geopotential

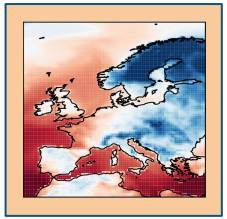
RCM
GCM

Large-scale variables
(Low-resolution)



Downscaling Function

y



Temperature
Precipitation

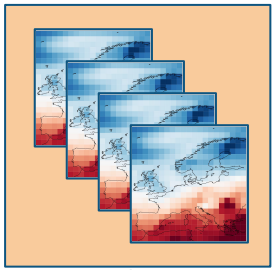
Surface variable
(High-resolution)

* RCM emulation generally focuses on local-scale variables (e.g., temperature and precipitation)

How are RCM emulators trained?

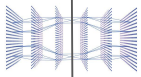
How can we emulate a RCM with DL? *

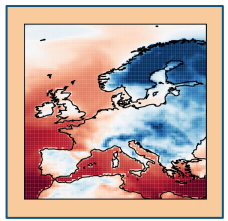
Large-scale variables
(Low-resolution)



$\mathbf{x}$

Air-temperature
Specific Humidity
Wind Components
Geopotential





y

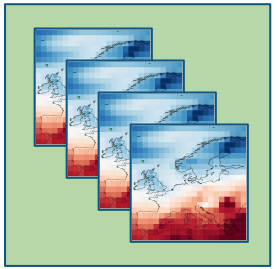
Temperature
Precipitation

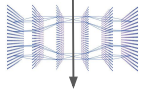
Surface variable
(High-resolution)

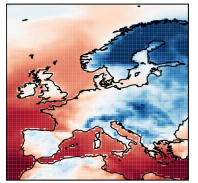
RCM
GCM

Downscaling Function

Large-scale variables
(Low-resolution)





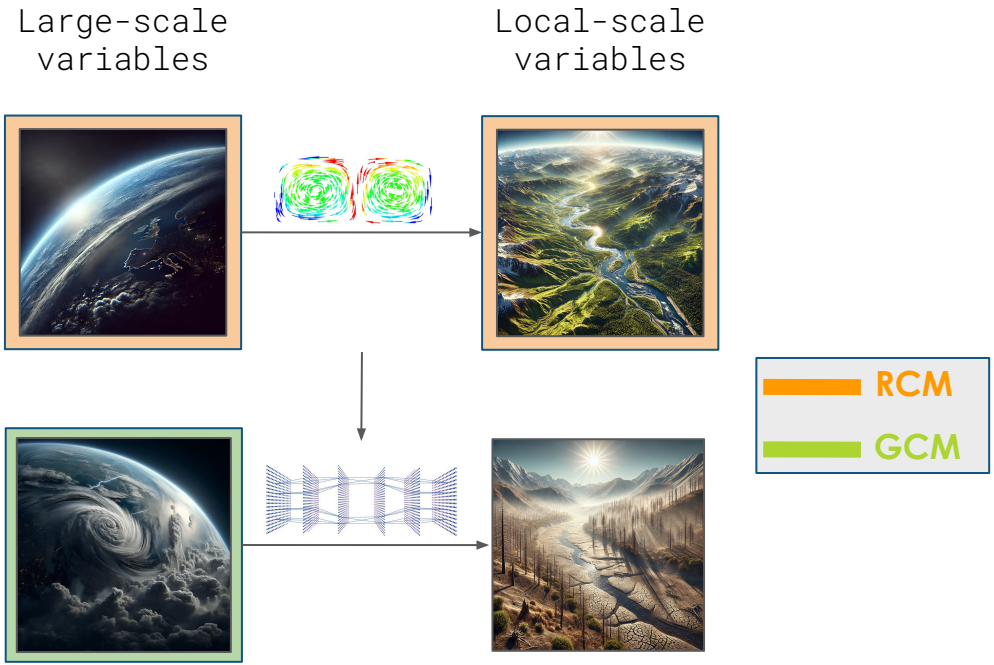


Emulated RCM
projection

* RCM emulation generally focuses on local-scale variables (e.g., temperature and precipitation)

Weather Turing Test

What are we actually learning from this approach to RCM emulation?



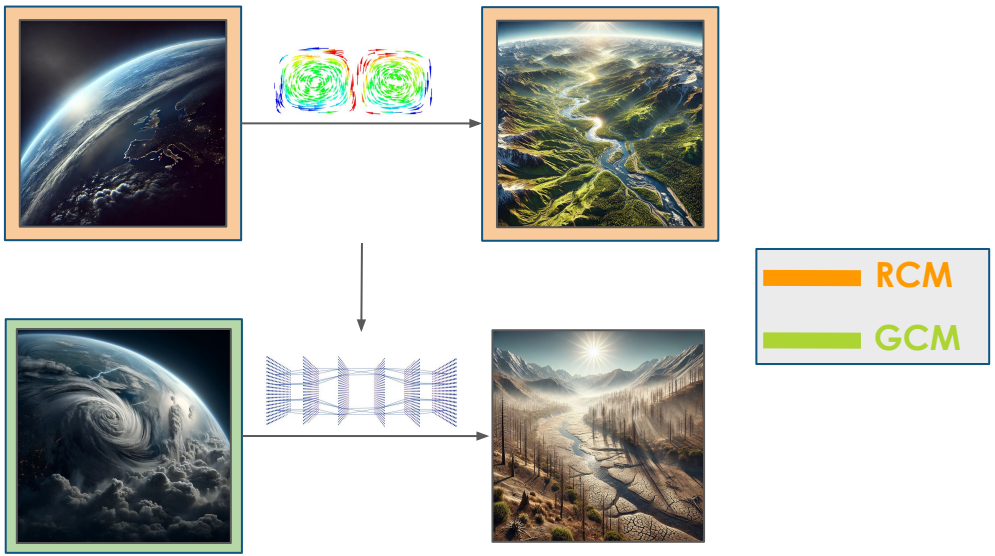
We are learning how the RCM **connects large-scale processes with local-scale dynamics.**

Weather Turing Test

What are we actually learning from this approach to RCM emulation?

Large-scale variables

Local-scale variables



We are learning how the RCM **connects large-scale processes with local-scale dynamics.**

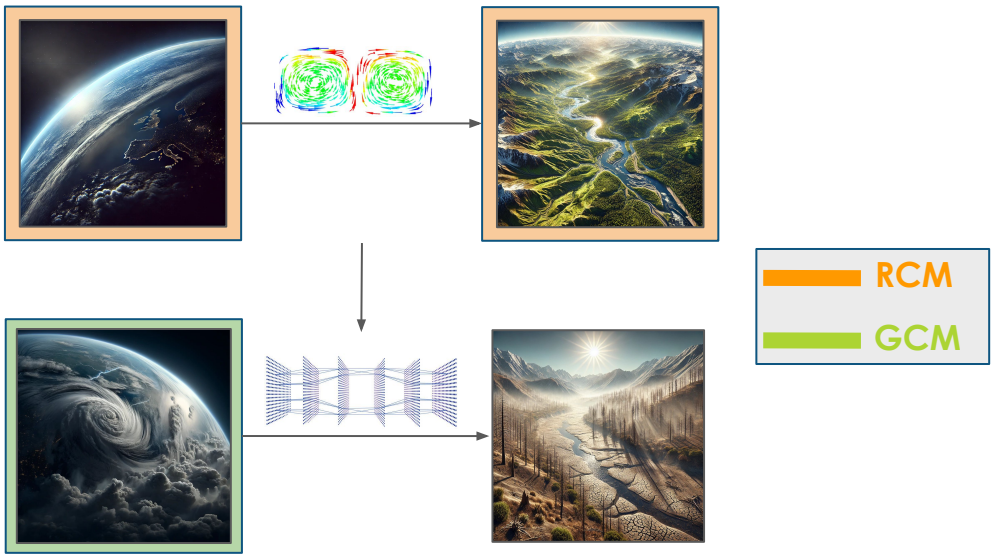
This allows us to obtain **high-resolution simulations for different GCMs** without running the RCM nested within each GCM.

Weather Turing Test

What are we actually learning from this approach to RCM emulation?

Large-scale variables

Local-scale variables



Why do we use large-scale variables as predictor?

We are learning how the RCM **connects large-scale processes with local-scale dynamics.**

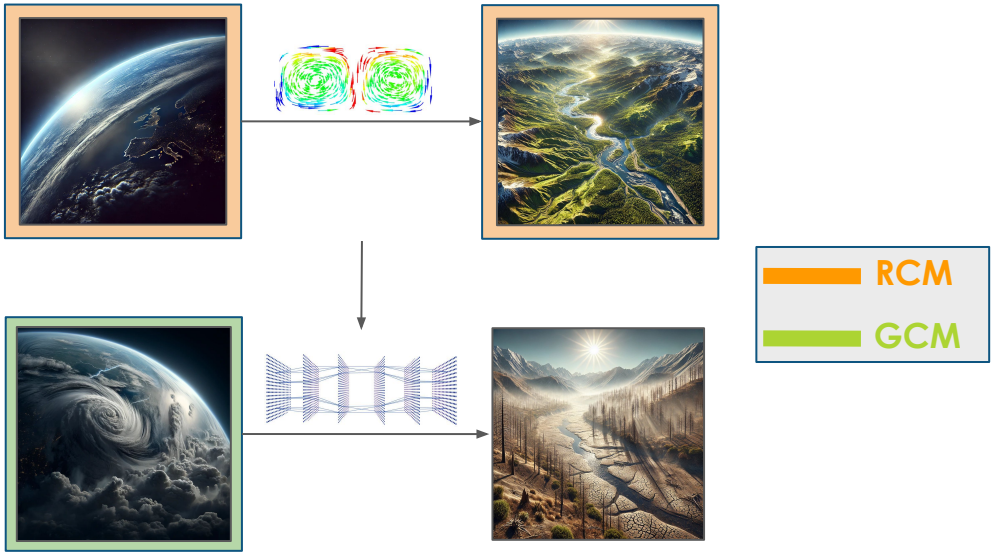
This allows us to obtain **high-resolution simulations for different GCMs** without running the RCM nested within each GCM.

Weather Turing Test

What are we actually learning from this approach to RCM emulation?

Large-scale variables

Local-scale variables



We are learning how the RCM **connects large-scale processes with local-scale dynamics.**

Why do we use large-scale variables as predictor?

The typical coarse resolution of these variables is sufficient to capture their key physical phenomena.

This allows us to obtain **high-resolution simulations for different GCMs** without running the RCM nested within each GCM.

Conclusions

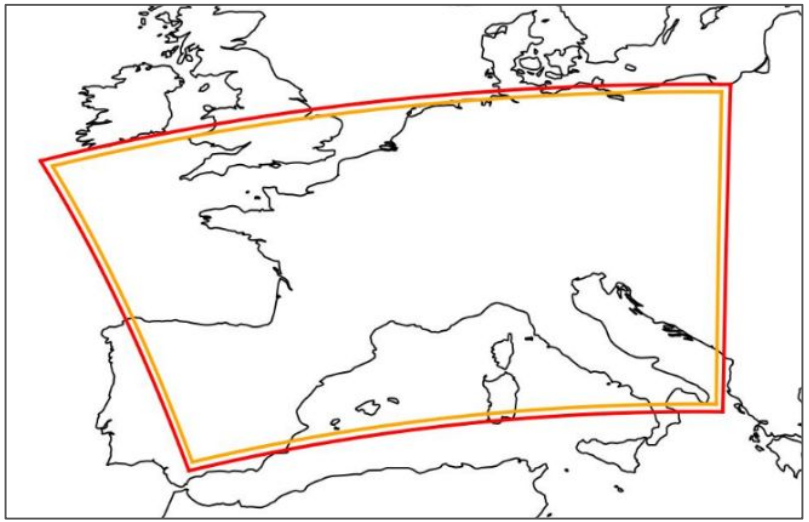
- Global Climate Models (**GCMs**) are physical models that simulate the Earth's climate under different future scenarios.
  - **GCMs** work well at large scales but **struggle to capture local-scale detail.**
  - To address this, Regional Climate Models (**RCMs**) are **nested within GCMs** and run at **higher resolution over specific regions**. However, RCMs are computationally expensive.

- Deep Learning (**DL**) models can emulate the downscaling process (from coarse large-scale to fine local-scale), thus **emulating the RCM**.
  - Once trained, these emulators can be **applied to different GCM outputs**, enabling **fast and cost-efficient generation of high-resolution climate simulations.**

Table of Contents

Region of Interest

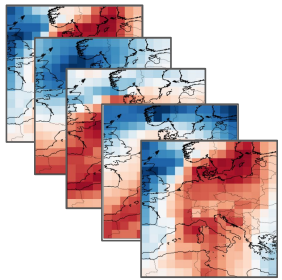
Suppose we need to study the **effects of climate change on future scenarios** for the **Alps region**.



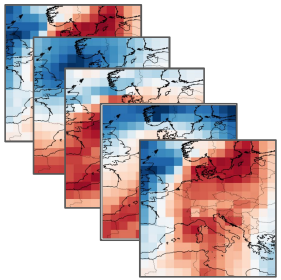
GCM Simulations

To do this, we obtain simulations of **future climate change scenarios from a GCM**, which we will denote as **GCM_1**.

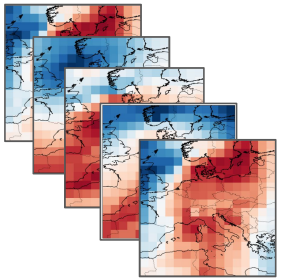
01-01-2080



02-01-2080



03-01-2080

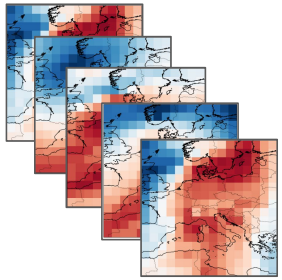


...

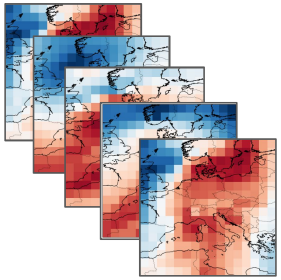
GCM Simulations

To do this, we obtain simulations of **future climate change scenarios from a GCM**, which we will denote as **GCM_1**.

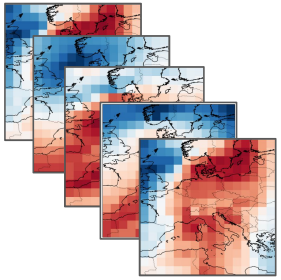
01-01-2080



02-01-2080

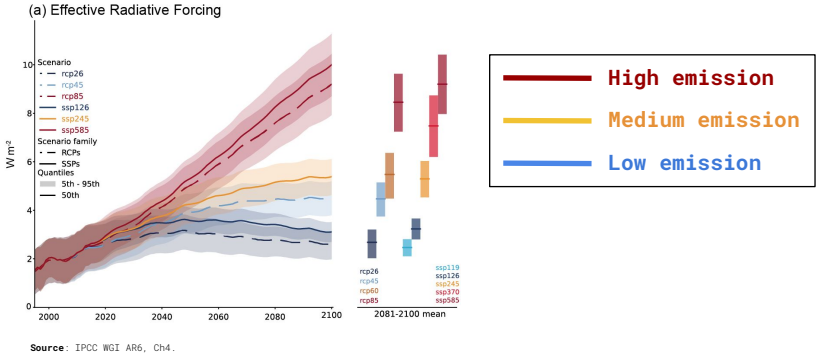


03-01-2080



...

When running future simulations, we assume specific **emissions scenarios**

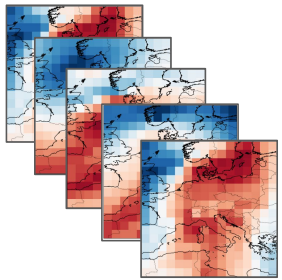


Source: IPCC WGI AR6, Ch4.

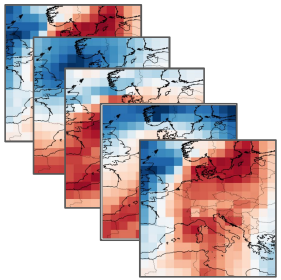
GCM Simulations

To do this, we obtain simulations of **future climate change scenarios from a GCM**, which we will denote as **GCM_1**.

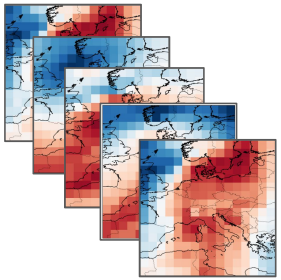
01-01-2080



02-01-2080

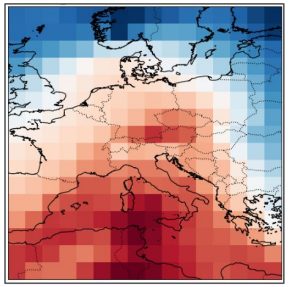


03-01-2080



...

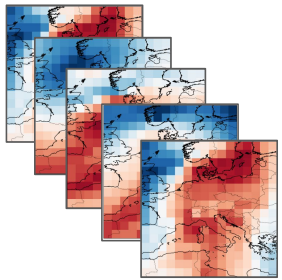
Unfortunately, due to the limitations of GCMs, the **simulations from GCM_1 cannot accurately reproduce local-scale conditions**



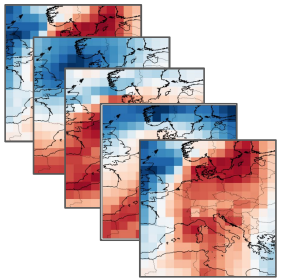
GCM Simulations

To do this, we obtain simulations of **future climate change scenarios from a GCM**, which we will denote as **GCM_1**.

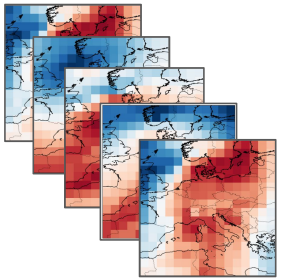
01-01-2080



02-01-2080

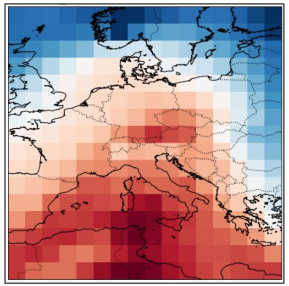


03-01-2080



...

Unfortunately, due to the limitations of GCMs, the **simulations from GCM_1 cannot accurately reproduce local-scale conditions**

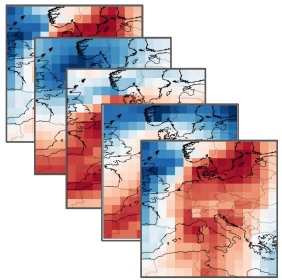


We need to run a **RCM**!

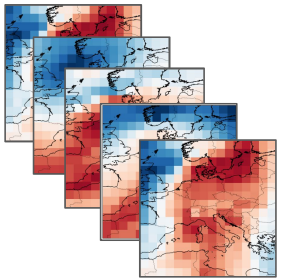
RCM Simulations

We run an RCM using the boundary conditions from GCM_1, thereby obtaining local-scale simulations of the future climate for the Alps region.

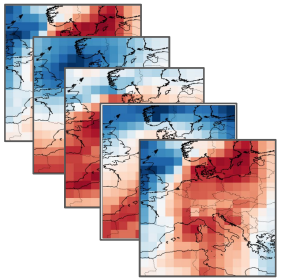
01-01-2080



02-01-2080



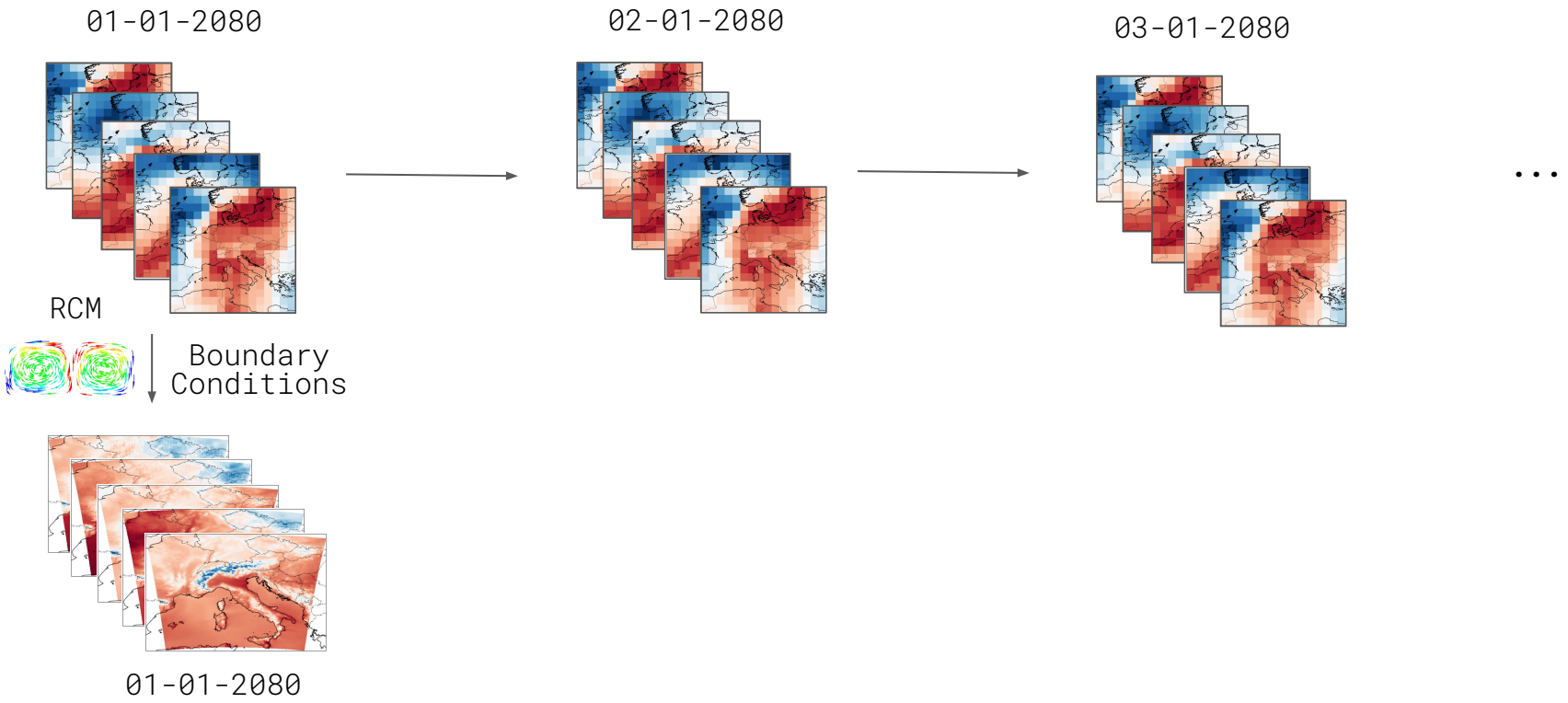
03-01-2080



...

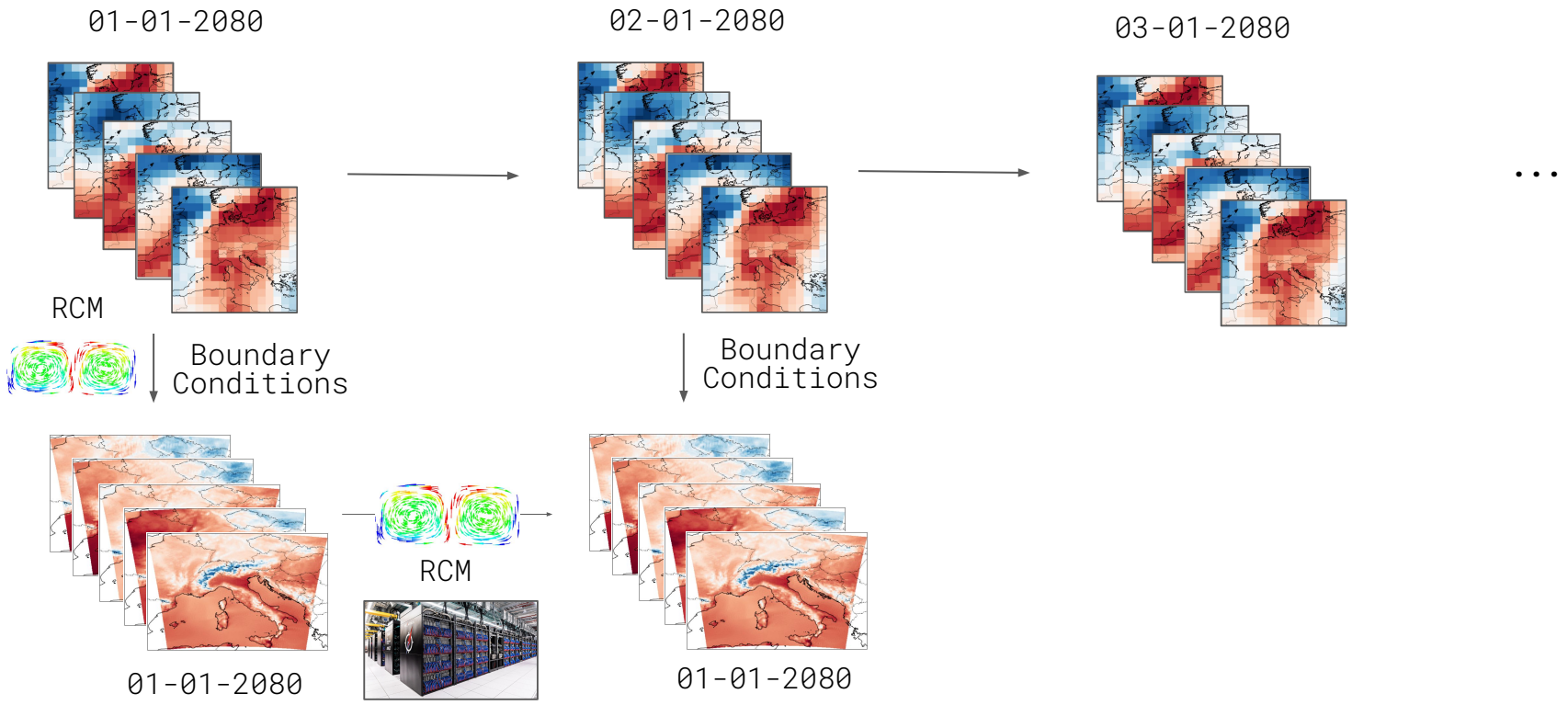
RCM Simulations

We run an RCM using the boundary conditions from GCM_1, thereby obtaining local-scale simulations of the future climate for the Alps region.



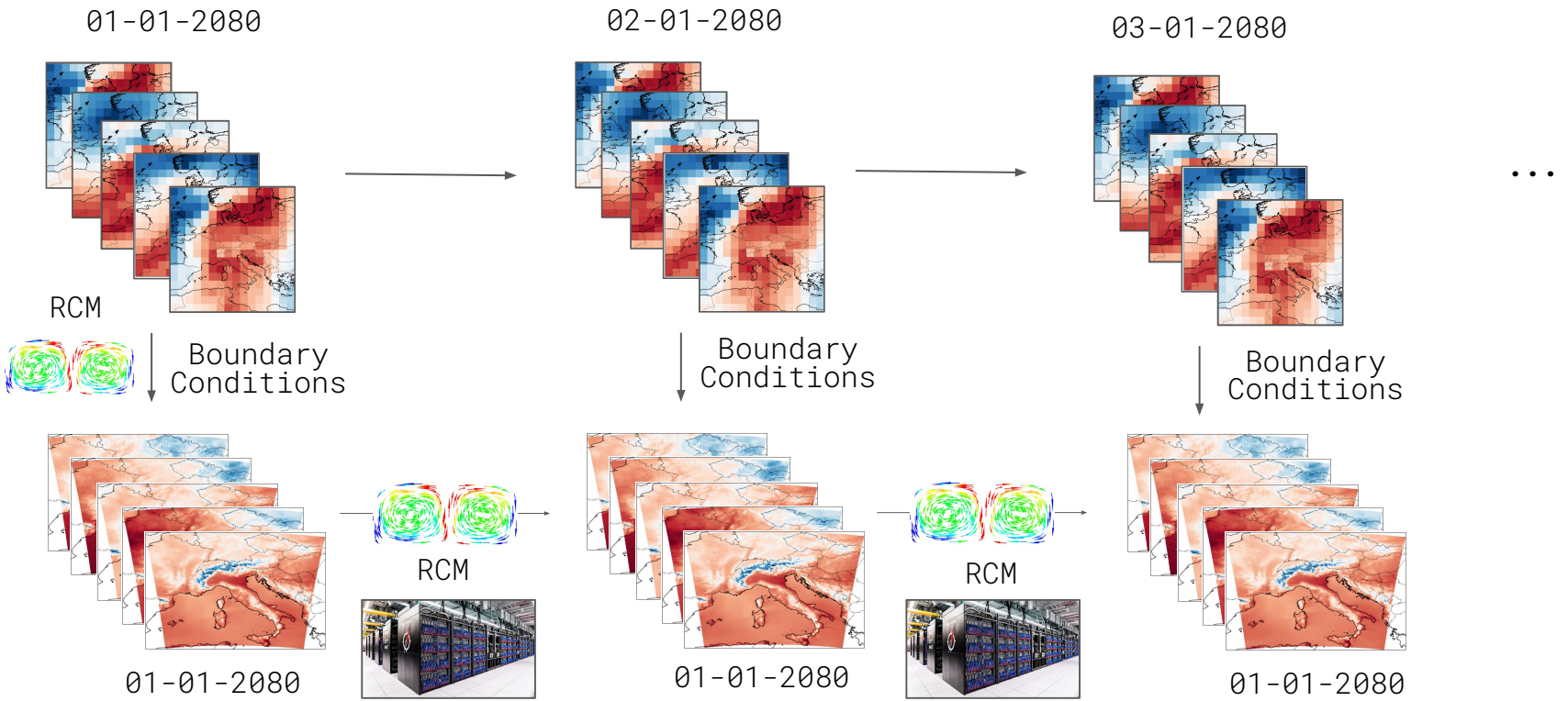
RCM Simulations

We run an RCM using the boundary conditions from GCM_1, thereby obtaining local-scale simulations of the future climate for the Alps region.



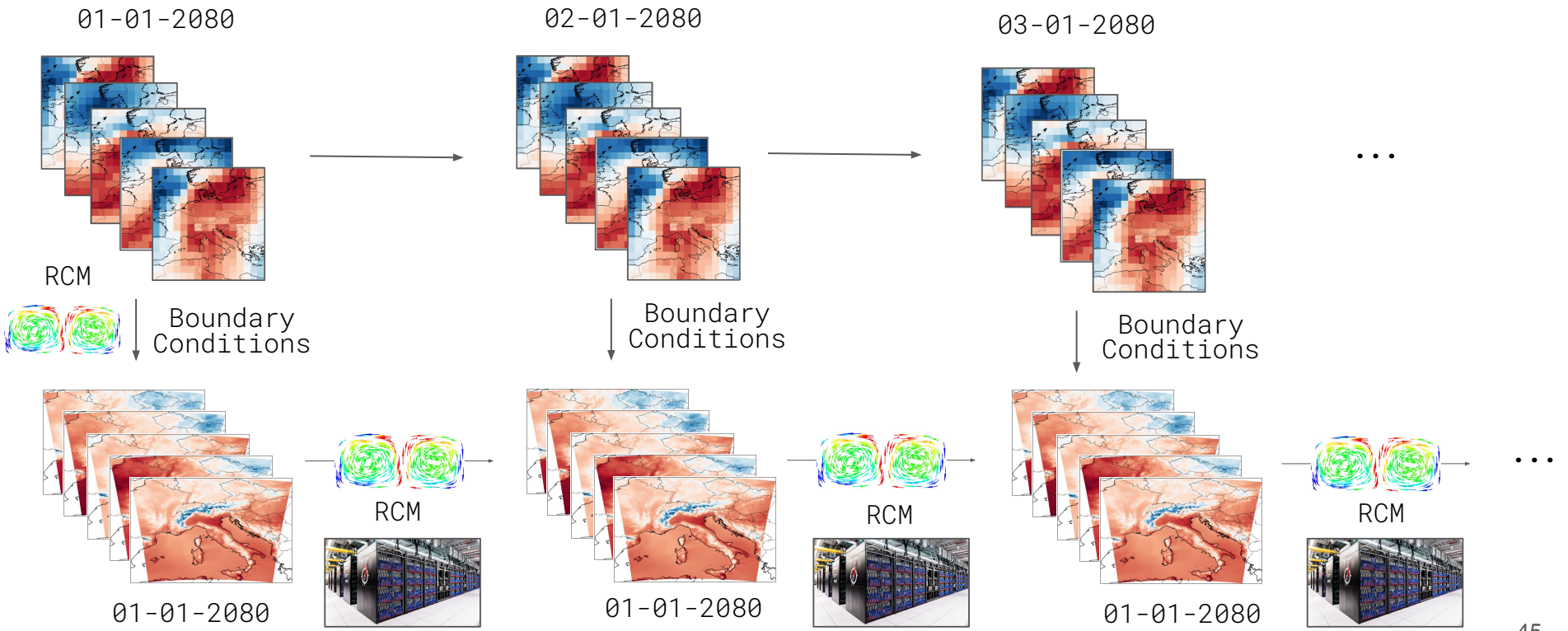
RCM Simulations

We run an RCM using the boundary conditions from GCM_1, thereby obtaining local-scale simulations of the future climate for the Alps region.



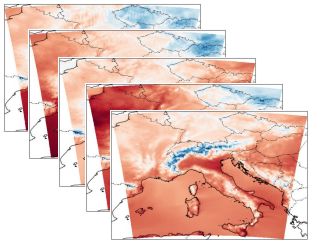
RCM Simulations

We run an RCM using the boundary conditions from GCM_1, thereby obtaining local-scale simulations of the future climate for the Alps region.

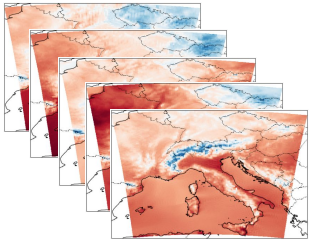


RCM Simulations

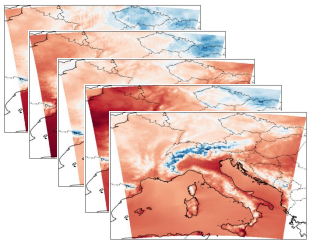
After **running the RCM** (with **GCM_1** as **boundary conditions**), we obtained local-scale simulations for the Alps region.



01-01-2080



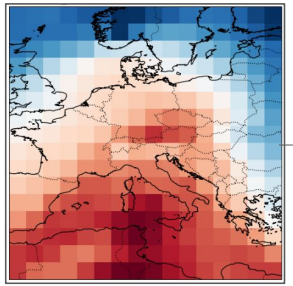
01-01-2080



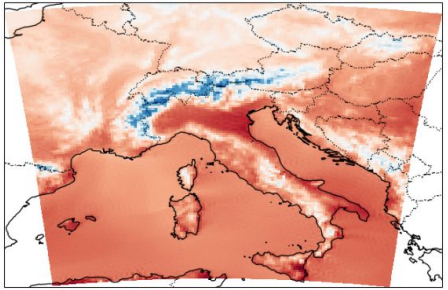
01-01-2080

...

GCM



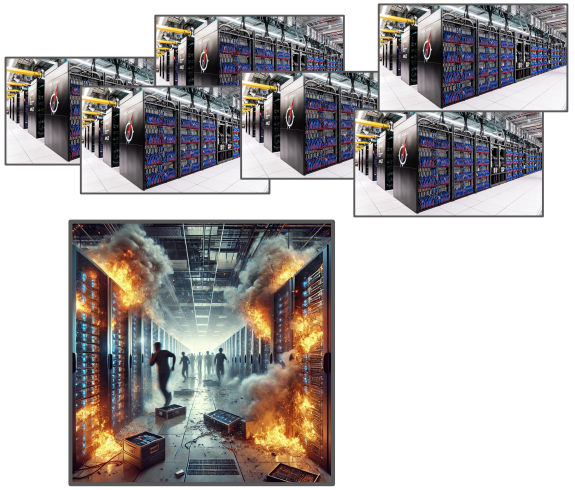
RCM



RCM Emulation

But what if we want to run the RCM with **other GCMs** or under **different future scenarios**?

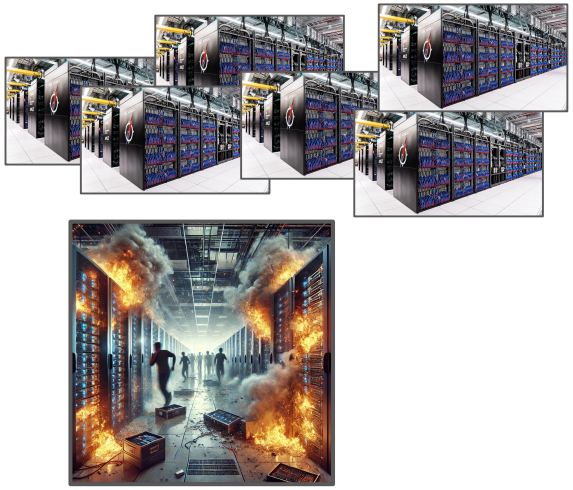
We need to re-run the RCM with different GCMs as boundary conditions → Computationally demanding



RCM Emulation

But what if we want to run the RCM with **other GCMs** or under **different future scenarios**?

We need to re-run the RCM with different GCMs as boundary conditions → Computationally demanding



RCM Emulation

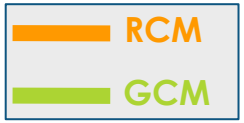
Deep Learning ~ RCM

Perfect and Imperfect Frameworks

Our objective is to **emulate the downscaling function of the RCM** (run using GCM_1 as boundary conditions) so that **it can be applied to different GCMs_X**, thereby generating the corresponding simulations.

Perfect Framework

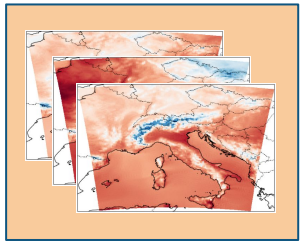
Imperfect Framework



Perfect and Imperfect Frameworks

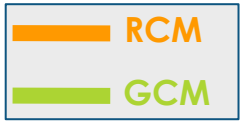
Our objective is to **emulate the downscaling function of the RCM** (run using GCM_1 as boundary conditions) so that **it can be applied to different GCMs_X**, thereby generating the corresponding simulations.

Perfect Framework



Large-scale variables

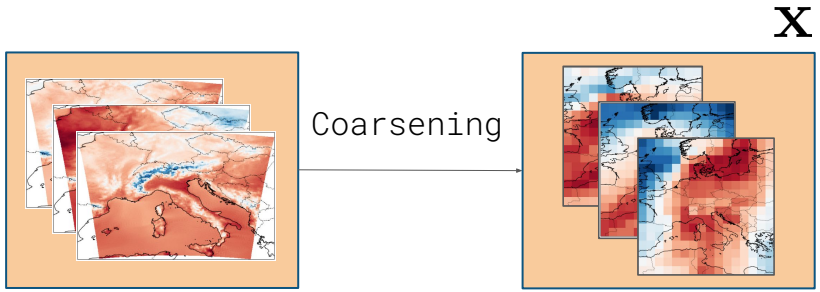
Imperfect Framework



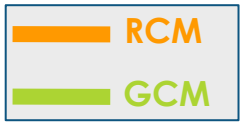
Perfect and Imperfect Frameworks

Our objective is to **emulate the downscaling function of the RCM** (run using GCM_1 as boundary conditions) so that **it can be applied to different GCMs_X**, thereby generating the corresponding simulations.

Perfect Framework



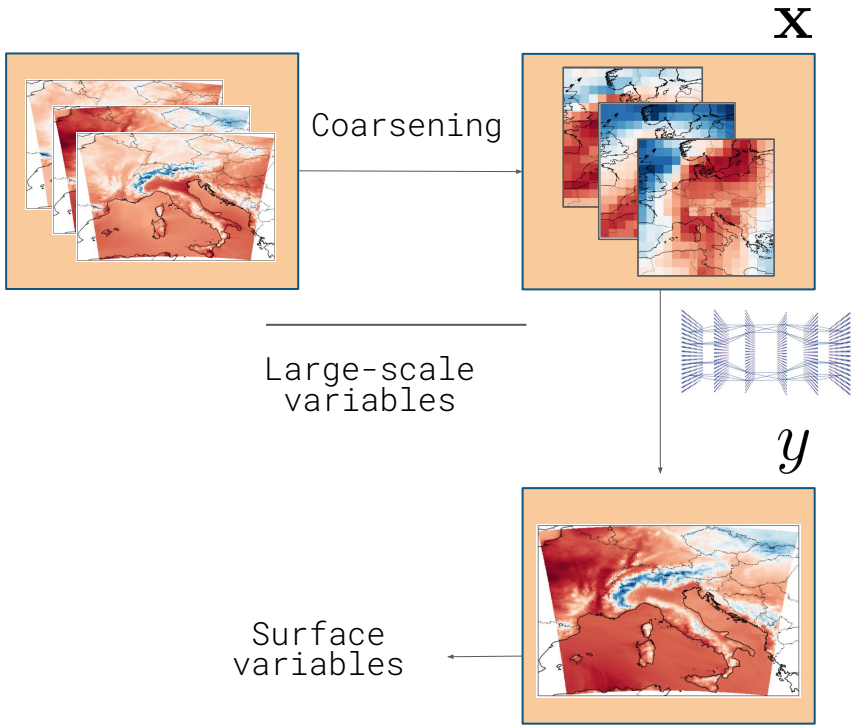
Imperfect Framework



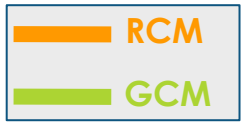
Perfect and Imperfect Frameworks

Our objective is to **emulate the downscaling function of the RCM** (run using GCM_1 as boundary conditions) so that **it can be applied to different GCMs_X**, thereby generating the corresponding simulations.

Perfect Framework



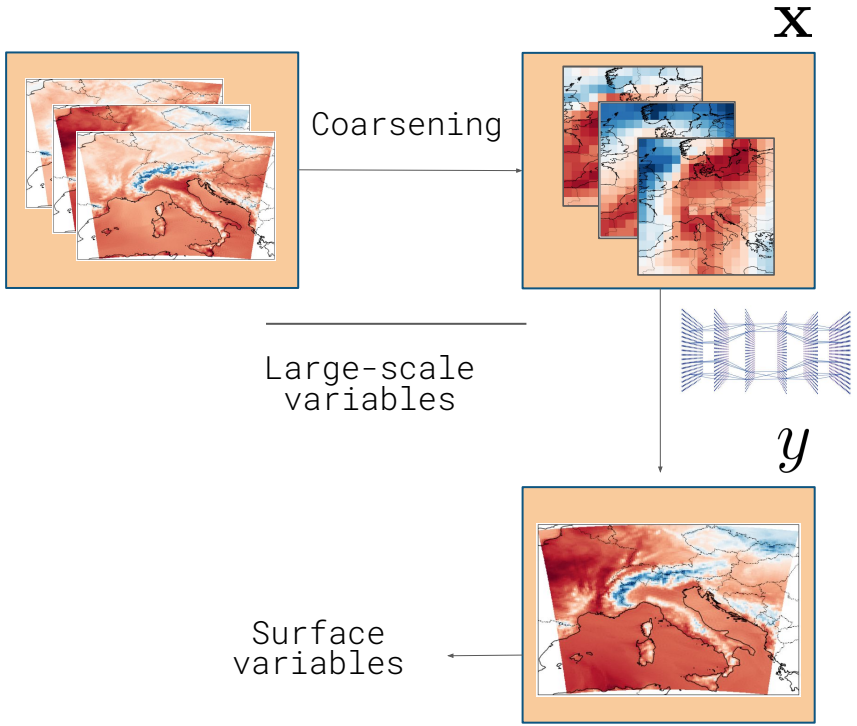
Imperfect Framework



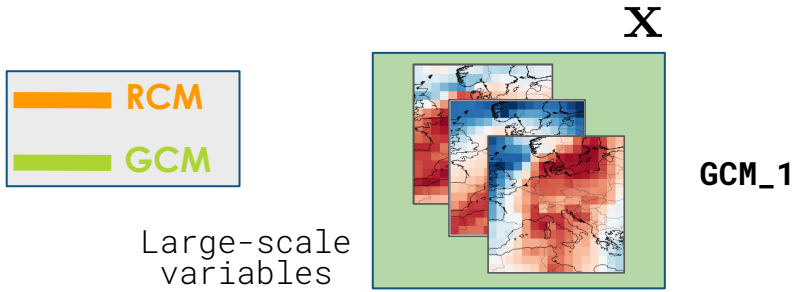
Perfect and Imperfect Frameworks

Our objective is to **emulate the downscaling function of the RCM** (run using GCM_1 as boundary conditions) so that **it can be applied to different GCMs_X**, thereby generating the corresponding simulations.

Perfect Framework



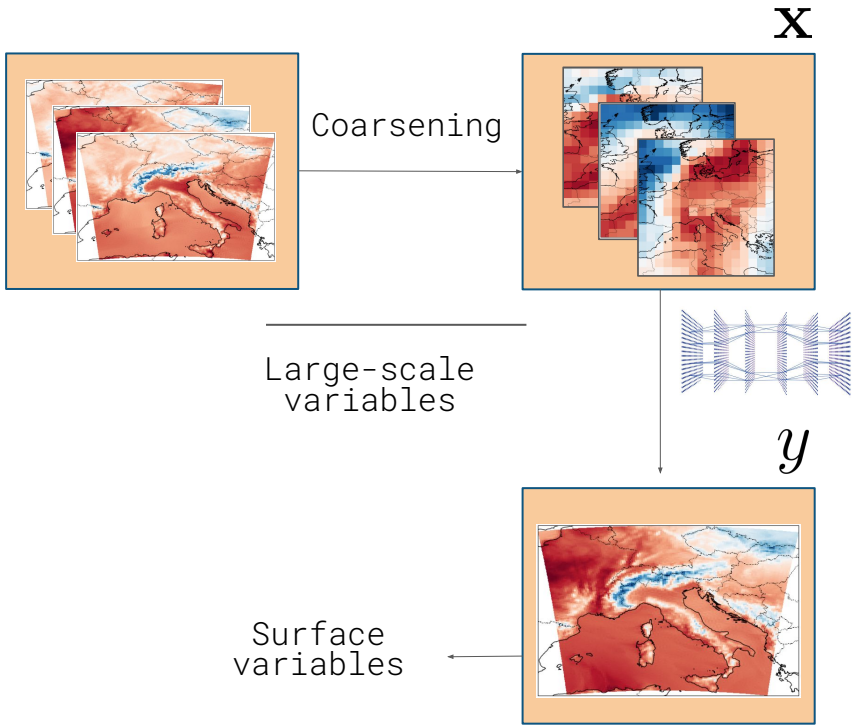
Imperfect Framework



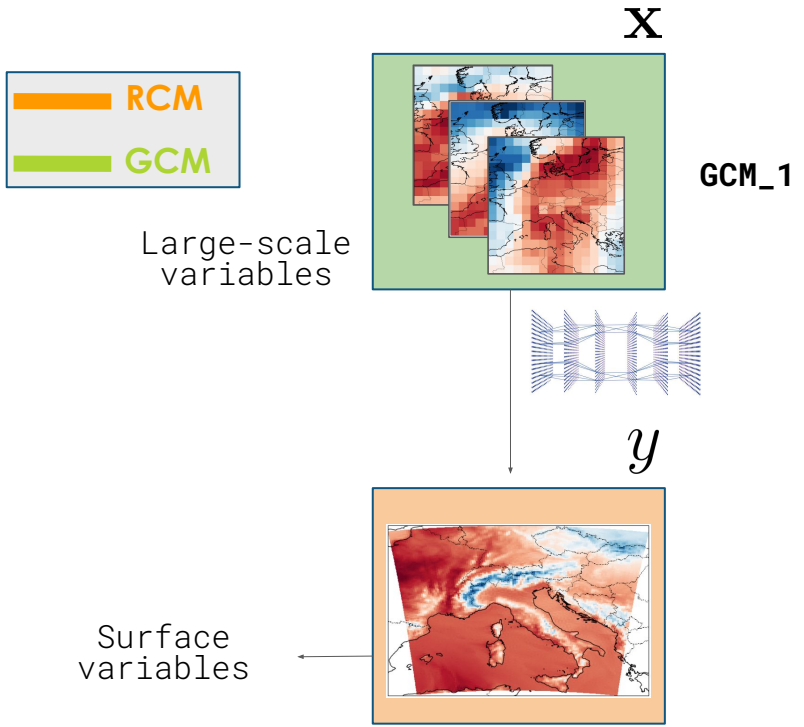
Perfect and Imperfect Frameworks

Our objective is to **emulate the downscaling function of the RCM** (run using GCM_1 as boundary conditions) so that **it can be applied to different GCMs_X**, thereby generating the corresponding simulations.

Perfect Framework

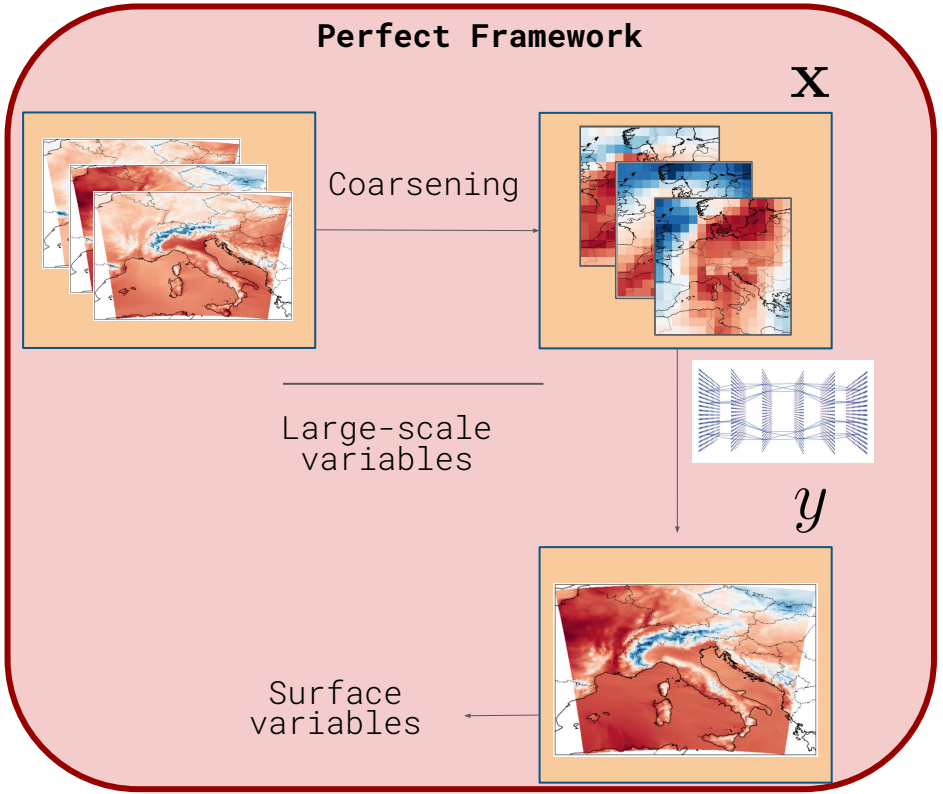


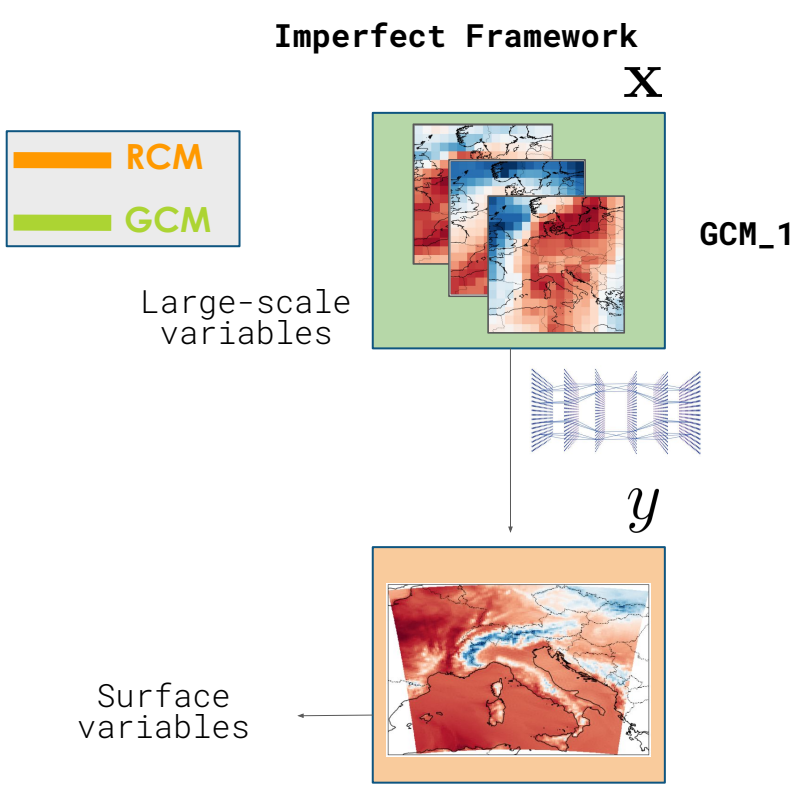
Imperfect Framework



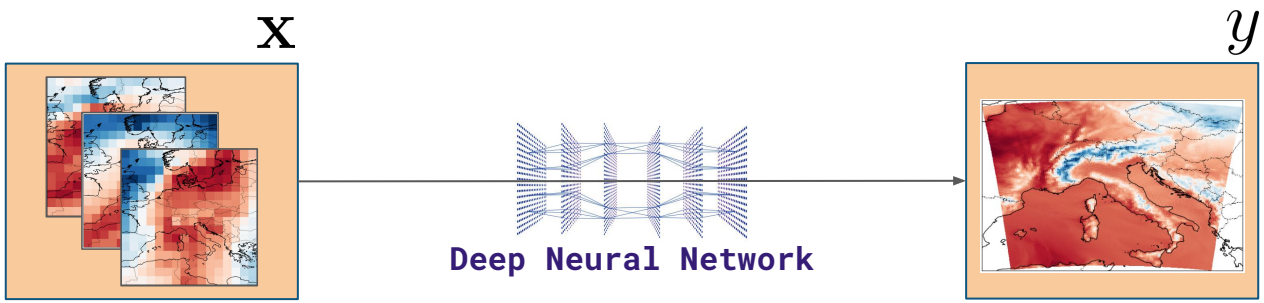
Perfect and Imperfect Frameworks

Our objective is to **emulate the downscaling function of the RCM** (run using GCM_1 as boundary conditions) so that **it can be applied to different GCMs_X**, thereby generating the corresponding simulations.



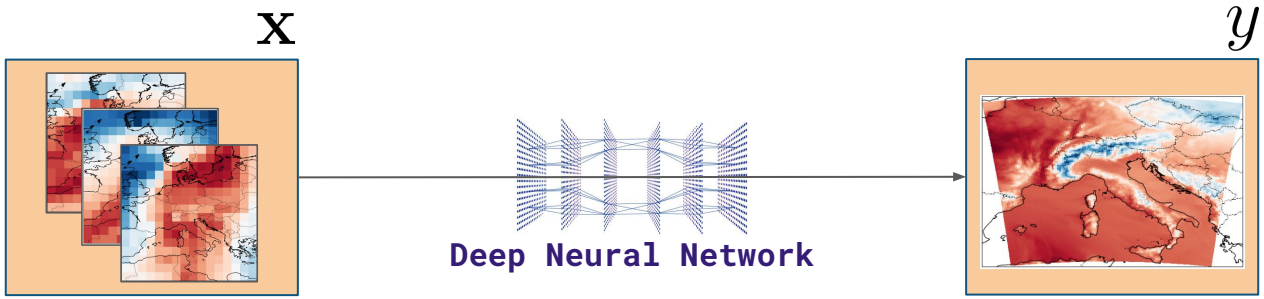


Training



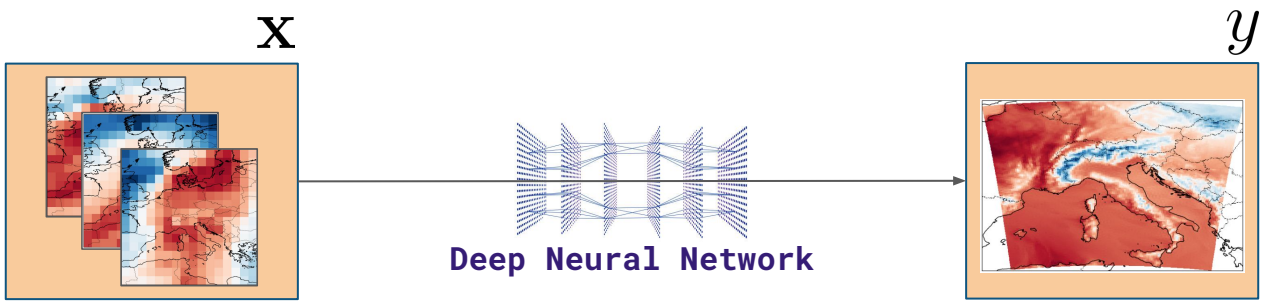
Full dataset: (1960-1980) and (2080-2100)

Training



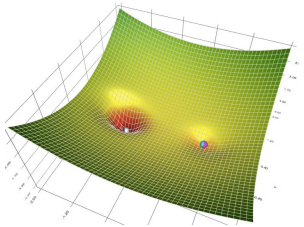
Full dataset: (1960-1980) and (2080-2100)

Training

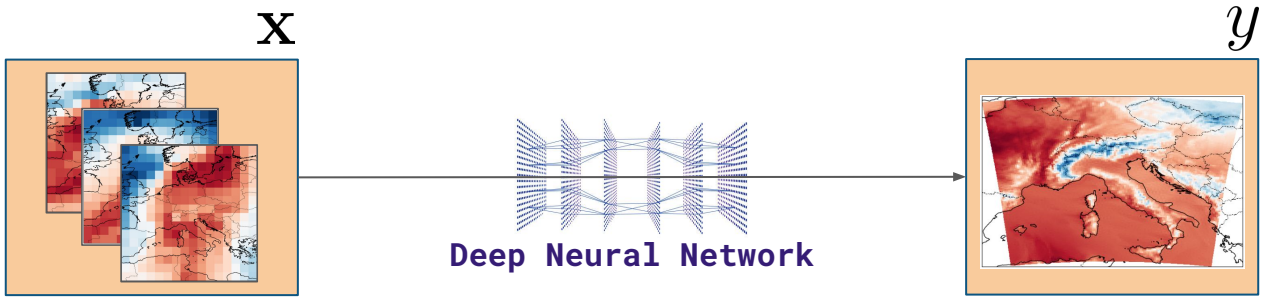


Full dataset: (1960-1980) and (2080-2100)

Training set: (1960-1975) and (2080-2095)



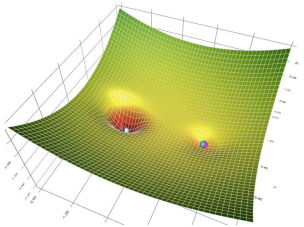
Training



Full dataset: (1960-1980) and (2080-2100)

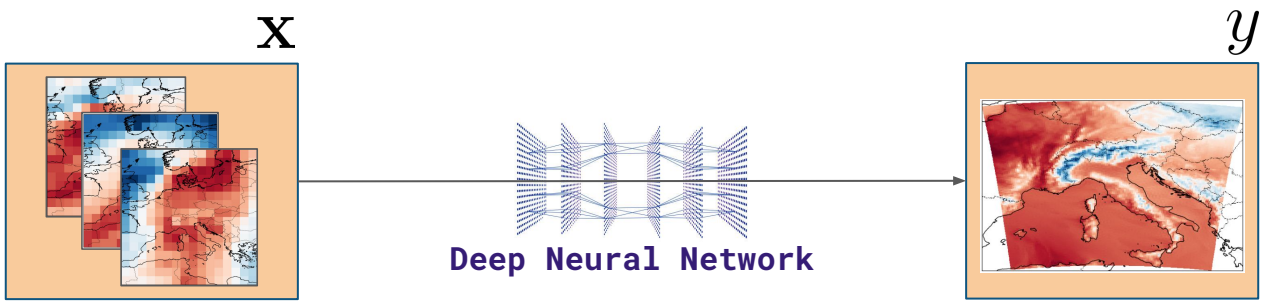
Training set: (1960-1975) and (2080-2095)

Test set: (1976-1980) and (2096-2100)

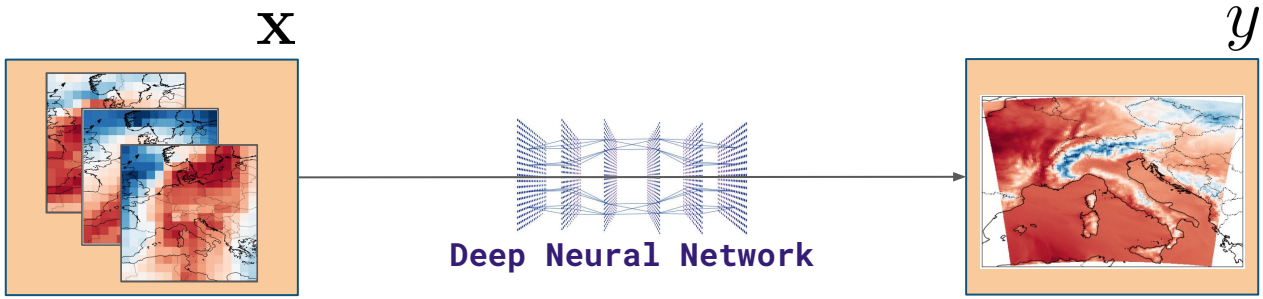


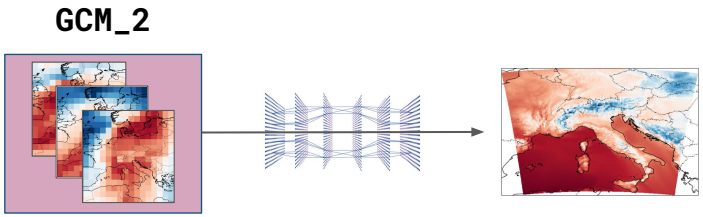
Is the model performing well on **unseen data?**

Inference

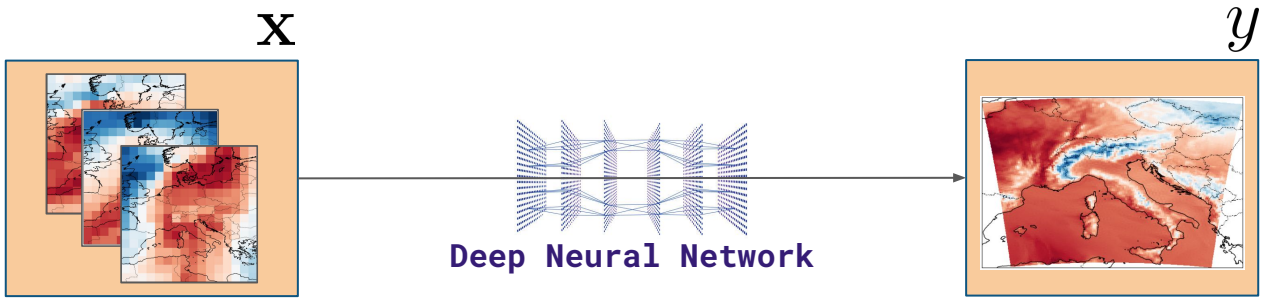


Inference

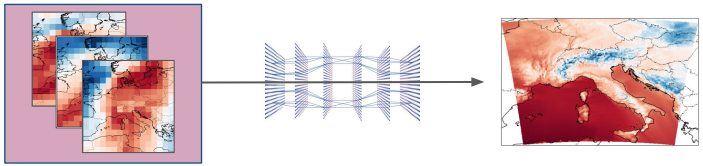




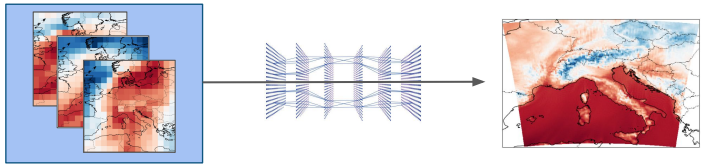
Inference



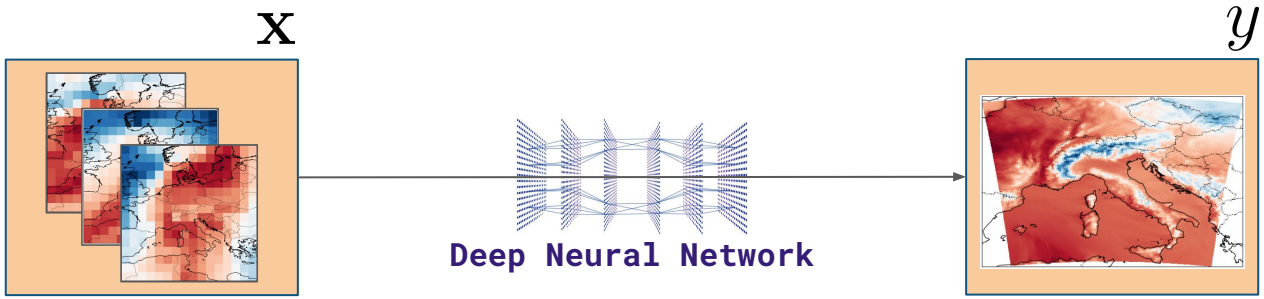
GCM_2

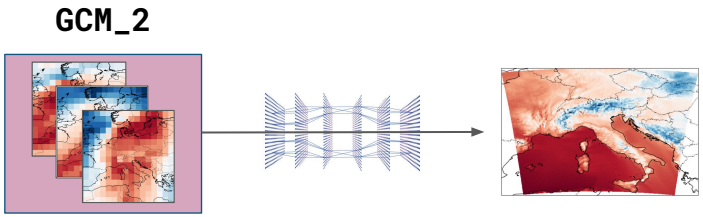


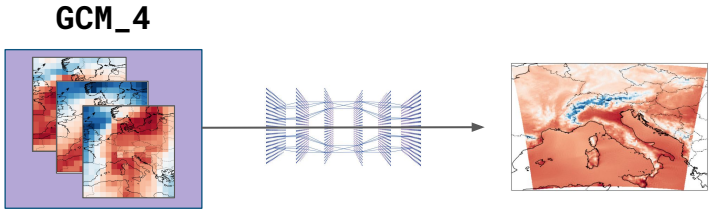
GCM_3

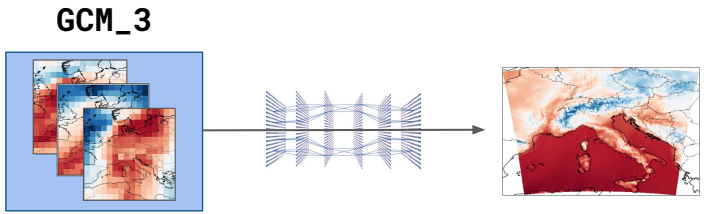


Inference

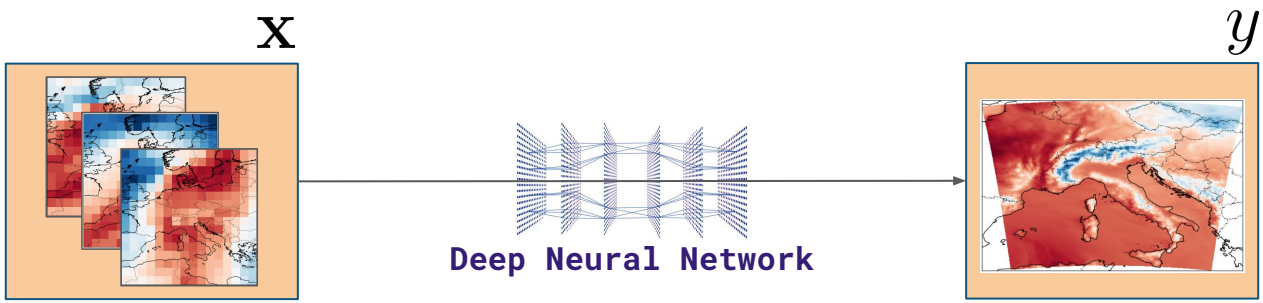




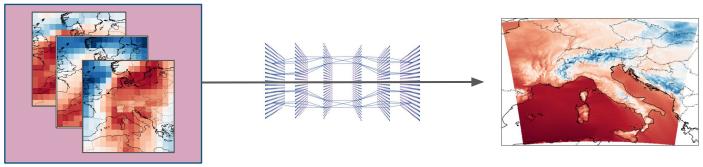




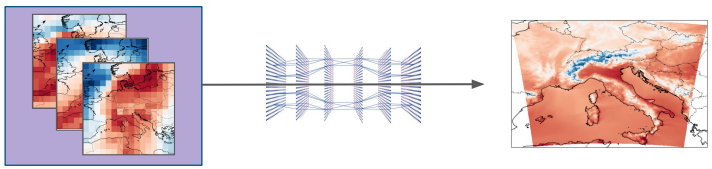
Inference



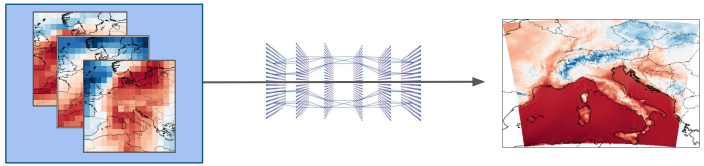
GCM_2



GCM_4



GCM_3



GCM_5

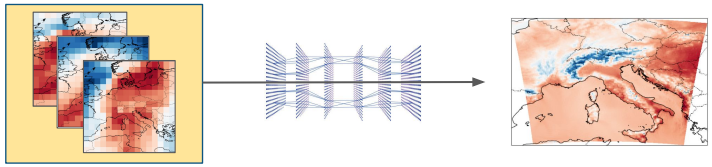
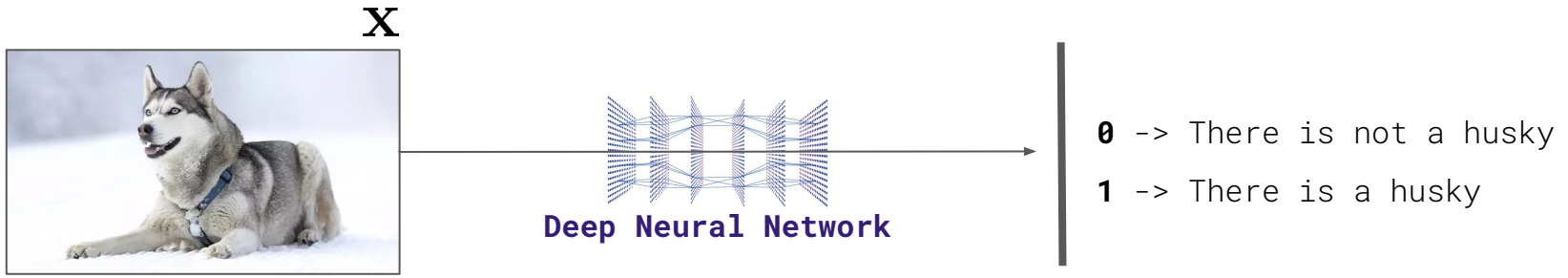


Table of Contents

Husky Classification

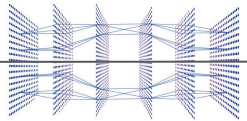
Suppose we want to train an **image classification model** to detect **Huskies** in images



Husky Classification

Suppose we want to train an **image classification model** to detect **Huskies** in images

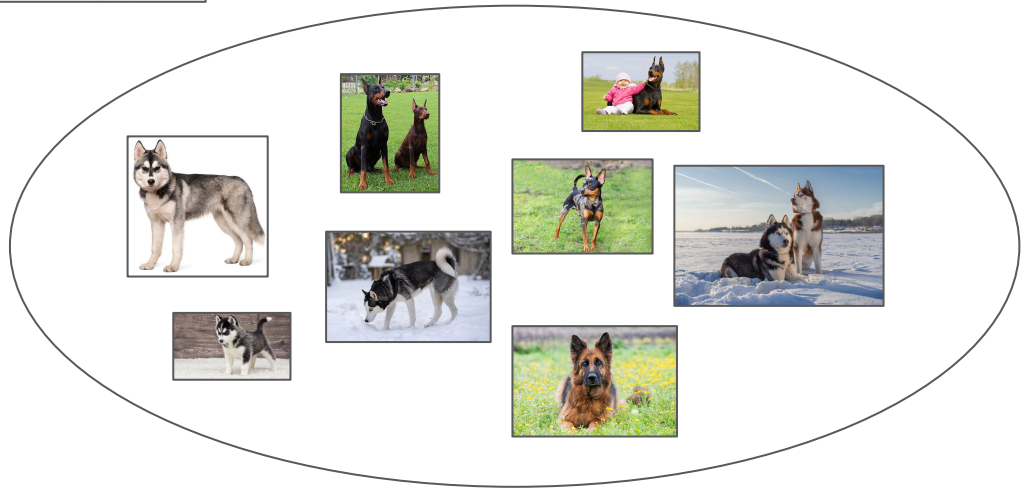
X



Deep Neural Network

0 -> There is not a husky

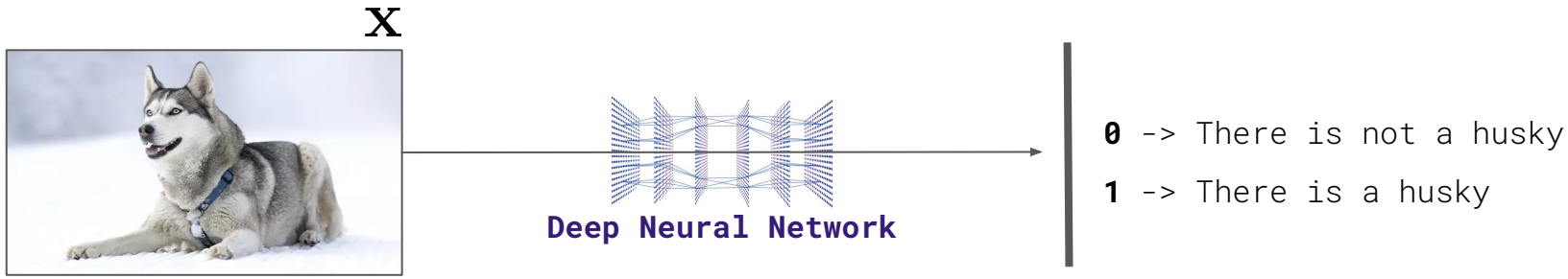
1 -> There is a husky



Training dataset

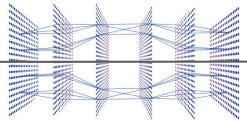
Husky Classification

To assess its **accuracy**, we evaluate it on a **test set**



Husky Classification

To assess its **accuracy**, we evaluate it on a **test set**



Deep Neural Network

0 -> There is not a husky

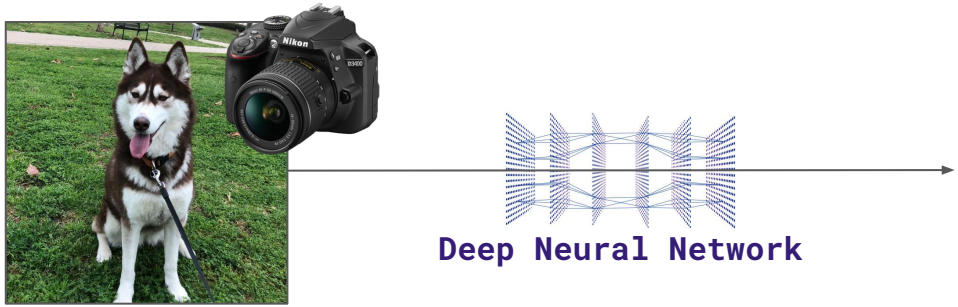
1 -> There is a husky

Test set → **Accuracy ~ 100%**

The model has learned the concept of a Husky, or has it?

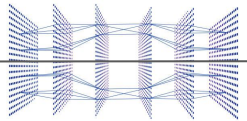
Husky Classification

However, when we test the model on **real-world data**, it **doesn't seem to work well**



Husky Classification

However, when we test the model on **real-world data**, it **doesn't seem to work well**

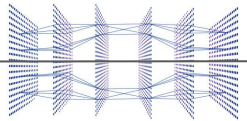


Deep Neural Network

0 -> There is not a husky

Husky Classification

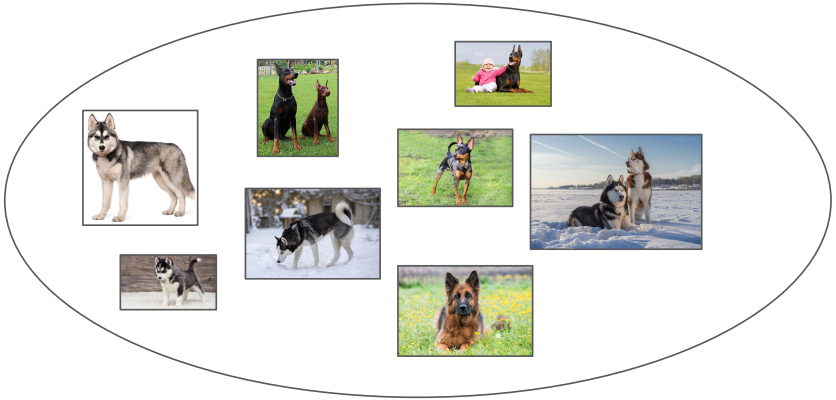
However, when we test the model on **real-world data**, it **doesn't seem to work well**

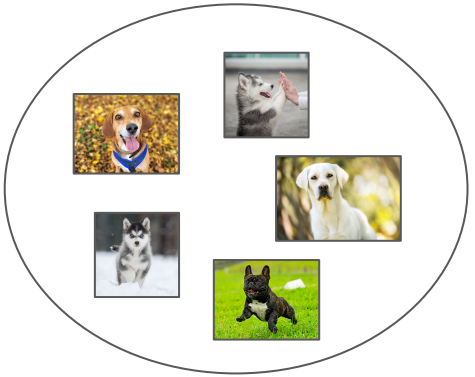


Deep Neural Network

0 -> There is not a husky

What is going on? Why is the model failing?





Husky Classification

The model is not **learning** the concept of Husky, but the **concept of snow**



Husky Classification

The model is not **learning** the concept of Husky, but the **concept of snow**



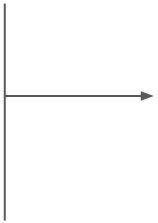
Evaluation goes beyond using a partition of the training distribution

Husky Classification

The model is not **learning** the concept of Husky, but the **concept of snow**



Evaluation goes beyond using a partition of the training distribution



Is the training distribution representative of the *target* distribution?

Husky Classification

The model is not **learning** the concept of Husky, but the **concept of snow**



Evaluation goes beyond using a partition of the training distribution

Is the training distribution representative of the *target* distribution?

Is the model learning the proper concepts (representations)? *

* Or is it relying on shortcut learning?

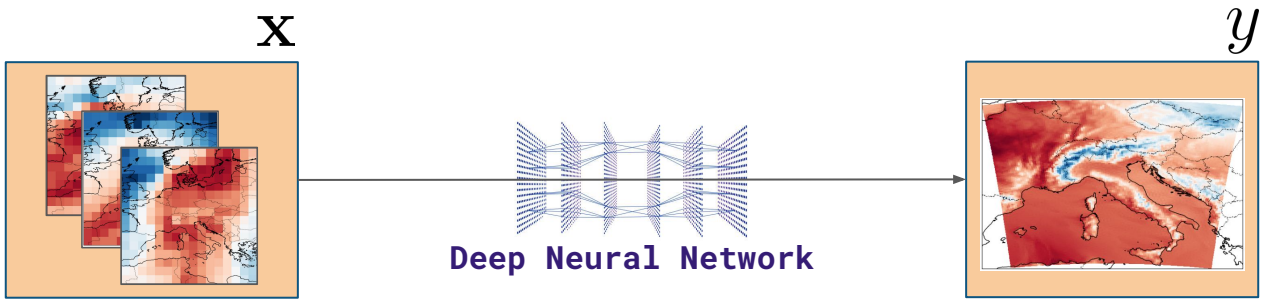
Geirhos, R., Jacobsen, J. H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11), 665-673.

What about RCM emulation?

Soft-transferability

What about RCM emulation?

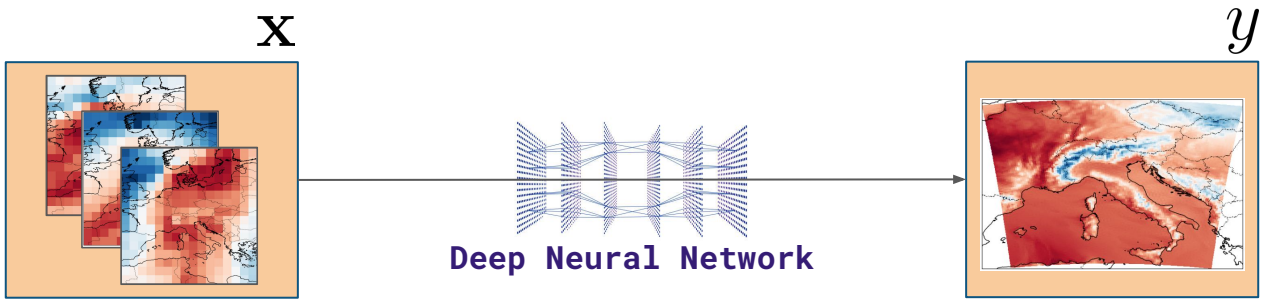
We have trained our DL model on a specific RCM ran with GCM_1 as boundary conditions



Full dataset: (1960-1980) and (2080-2100)

What about RCM emulation?

We have trained our DL model on a specific RCM ran with GCM_1 as boundary conditions

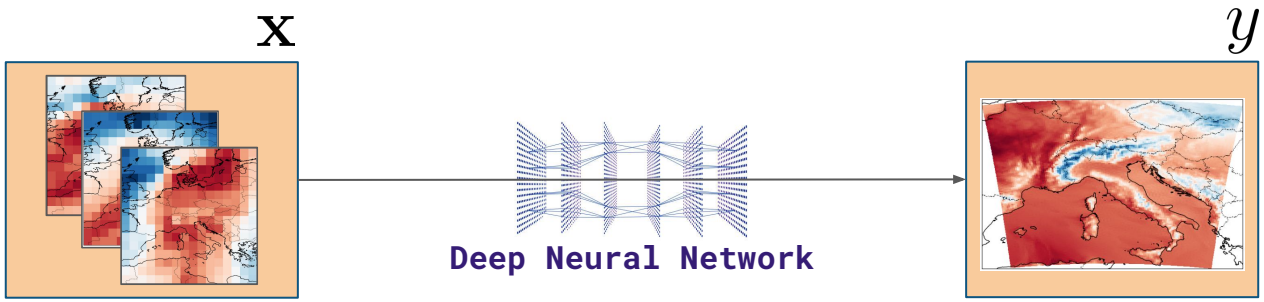


Full dataset: (1960-1980) and (2080-2100)

1. What if we simulate a **period not covered by the full dataset**?
2. What if we simulate an **emissions scenario different from that of GCM_1** used to run the RCM?

What about RCM emulation?

We have trained our DL model on a specific RCM ran with GCM_1 as boundary conditions



Full dataset: (1960-1980) and (2080-2100)

1. What if we simulate a **period not covered by the full dataset**?
2. What if we simulate an **emissions scenario different from that of GCM_1** used to run the RCM?

Soft-transferability

Interpolation/Extrapolation

What if we simulate a **period not covered by the full dataset**?

This is not a major issue for **RCM emulation**, since **we can train the model using both historical and future data**, allowing us to just having to evaluate its performance for **interpolation**.

Full dataset: (1960-1980) and (2080-2100)

Interpolation/Extrapolation

What if we simulate a **period not covered by the full dataset**?

This is not a major issue for **RCM emulation**, since **we can train the model using both historical and future data**, allowing us to just having to evaluate its performance for **interpolation**.

Full dataset: (1960-1980) and (2080-2100)

However, for other approaches like **observational statistical downscaling**, where **we have only historical data**, it is important to properly **evaluate how well they can extrapolate** to future scenarios.

OCTOBER 2025 GONZÁLEZ-ABAD AND GUTIÉRREZ

Are Deep Learning Methods Suitable for Downscaling Global Climate Projections? An Intercomparison for Temperature and Precipitation over Spain

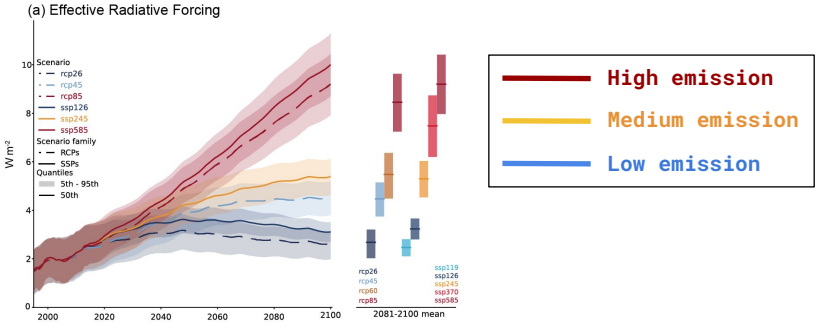
JOSE GONZÁLEZ-ABAD[a] AND JOSÉ MANUEL GUTIÉRREZ[a]

[a] *Instituto de Física de Cantabria (IFCA), CSIC-Universidad de Cantabria, Santander, Spain*

Different Scenarios

What if we simulate an **emissions scenario different from that of GCM_1** used to run the RCM?

Suppose the RCM simulations used for training were run assuming a **low emissions scenario**



What if I run the **inference on GCM_1**, but this time it was run under a **high emissions scenario?**

→ **Is the RCM emulator able to extrapolate to climates not seen during training?**

Different Scenarios

What if we simulate an **emissions scenario different from that of GCM_1** used to run the RCM?

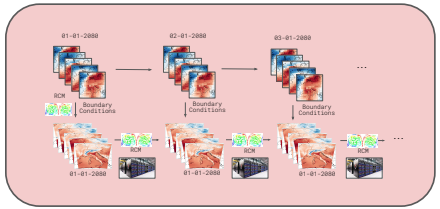
Assessing this property is key if we want to fill the GCM/RCM matrix across different scenarios

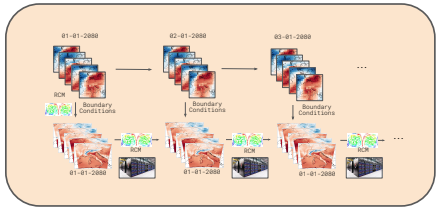
Different Scenarios

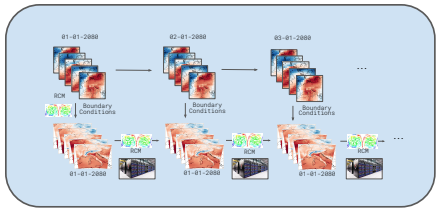
What if we simulate an **emissions scenario different from that of GCM_1** used to run the RCM?

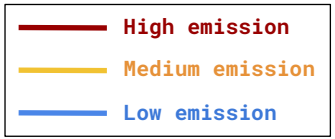
Assessing this property is key if we want to fill the GCM/RCM matrix across different scenarios

For this evaluation, we need **RCM simulations** that use GCM_1 as boundary conditions, **with different emissions scenarios**







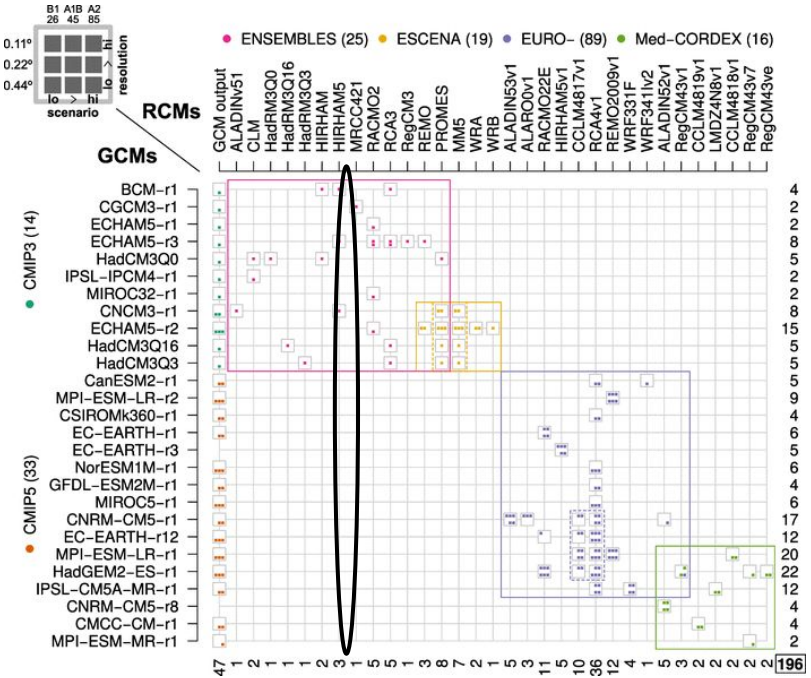


Different GCMs

Hard-transferability

Different GCMs

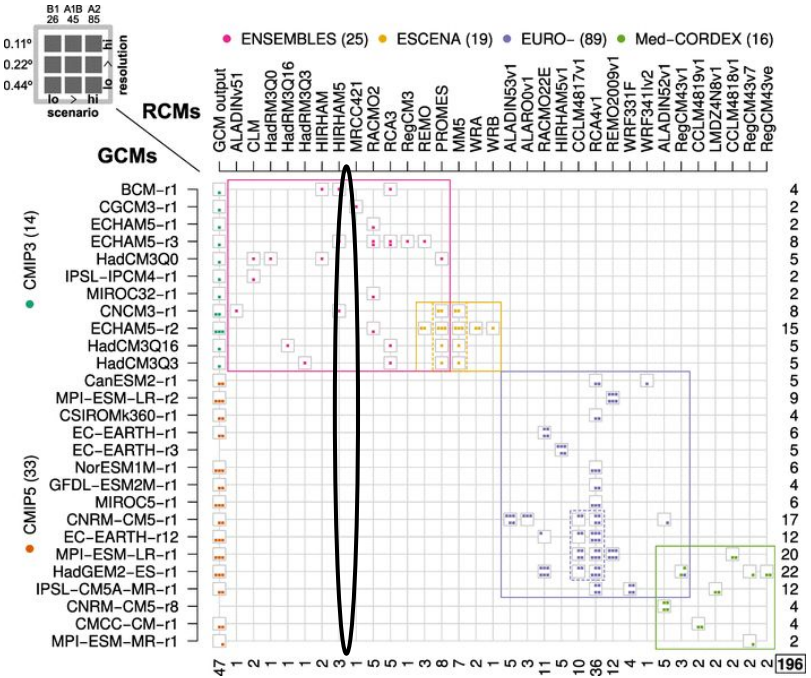
Our RCM emulator has been trained with simulations of an RCM run with GCM_1; however, **the main purpose of an RCM emulator is to generate simulations for different GCMs**



Fuente: Fernández, J., Frías, M. D., Cabos, W. D., Cofiño, A. S., Domínguez, M., Fita, L., ... & Sánchez, E. (2019). Consistency of climate change projections from multiple global and regional model intercomparison projects. Climate dynamics, 52, 1139-1156.

Different GCMs

Our RCM emulator has been trained with simulations of an RCM run with GCM_1; however, **the main purpose of an RCM emulator is to generate simulations for different GCMs**



Fuente: Fernández, J., Frías, M. D., Cabos, W. D., Cofiño, A. S., Domínguez, M., Fita, L., ... & Sánchez, E. (2019). Consistency of climate change projections from multiple global and regional model intercomparison projects. Climate dynamics, 52, 1139-1156.

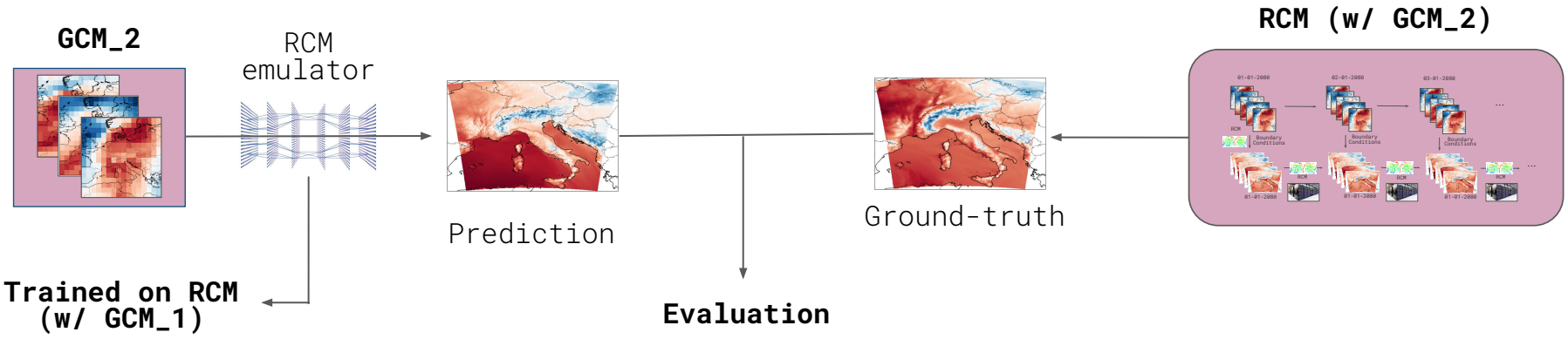
How can we evaluate the **performance** of the RCM emulator **across different GCMs**?

↓

Hard-transferability

Different GCMs

To evaluate the **hard-transferability** of a RCM emulator we need to run that same RCM but over a **different GCM** (for instance **GCM_2**)



Is the RCM emulator able to extrapolate to other GCMs?

Recap

Soft-transferability

Is the RCM emulator able to interpolate/extrapolate in time?

Is the RCM emulator able to extrapolate to climates (emission scenarios) not seen during training?

Hard-transferability

Is the RCM emulator able to extrapolate to other GCMs?

Recap

Soft-transferability

Is the RCM emulator able to interpolate/extrapolate in time?

Is the RCM emulator able to extrapolate to climates (emission scenarios) not seen during training?

Hard-transferability

Is the RCM emulator able to extrapolate to other GCMs?

Unfortunately, most evaluations require **re-running the RCM we aim to emulate**, which is precisely what RCM emulation seeks to avoid

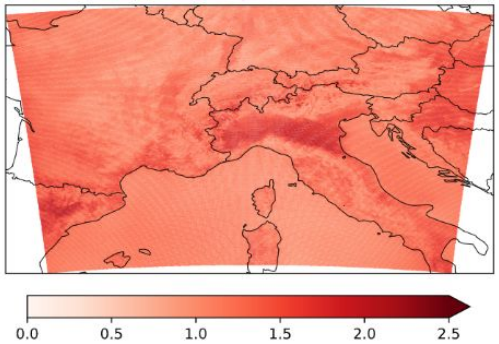
→ The proper evaluation of RCM emulator is challenging

ML-based Metrics

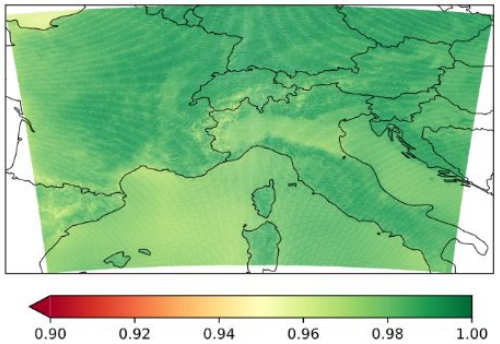
So far, we have reviewed the evaluation frameworks (i.e., what data to use for assessing different aspects of the RCM emulator), but **what metrics should we compute**?

Classic ML metrics (point-based)

Root Mean Squared Error (RMSE)



Pearson Correlation



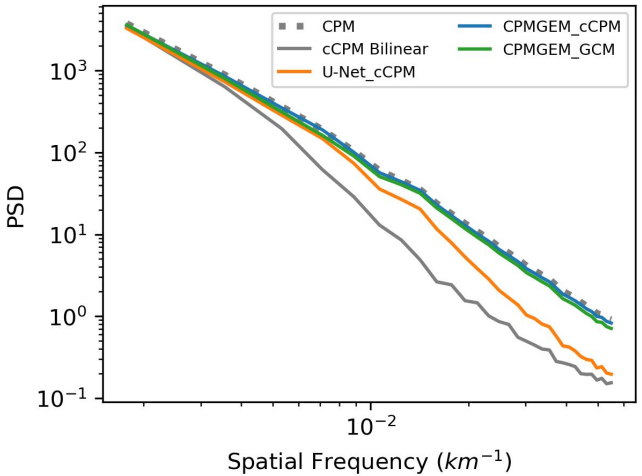
Ground-truth: RCM
Prediction: RCM emulator

ML-based Metrics

So far, we have reviewed the evaluation frameworks (i.e., what data to use for assessing different aspects of the RCM emulator), but **what metrics should we compute**?

Classic ML metrics

Power Spectral Density (PSD)



Is the RCM emulator producing **smoothed/blurry** predictions?

Ground-truth: RCM
Prediction: RCM emulator

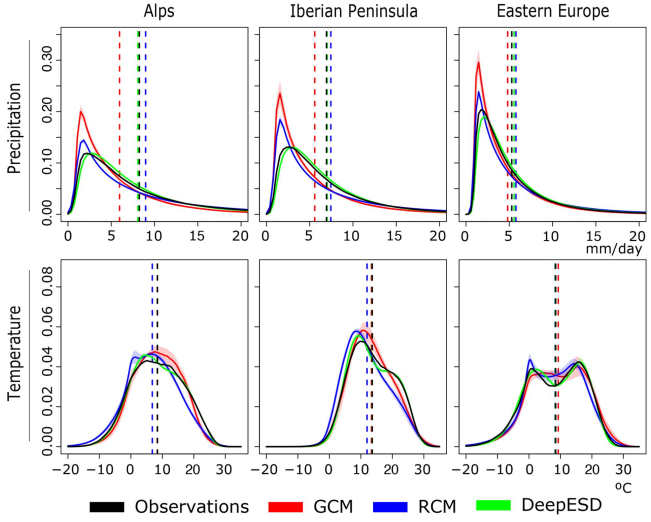
Source: Addison, H., Kendon, E., Ravuri, S., Aitchison, L., & Watson, P. A. (2024). Machine learning emulation of precipitation from km-scale regional climate simulations using a diffusion model. arXiv preprint arXiv:2407.14158.

Distributional-based Metrics

So far, we have reviewed the evaluation frameworks (i.e., what data to use for assessing different aspects of the RCM emulator), but **what metrics should we compute**?

Distributional-based metrics

The purpose of this set of metrics is to assess **how well the emulator reproduces the distribution of the RCM**



Source: Baño-Medina, J., Manzanas, R., Cimadevilla, E., Fernández, J., González-Abad, J., Cofiño, A. S., & Gutiérrez, J. M. (2022). Downscaling multi-model climate projection ensembles with deep learning (DeepESD): contribution to CORDEX EUR-44. Geoscientific Model Development Discussions, 2022, 1-14.

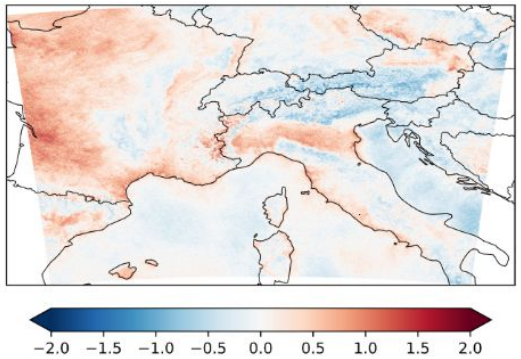
Ground-truth: RCM
Prediction: RCM emulator

Distributional-based Metrics

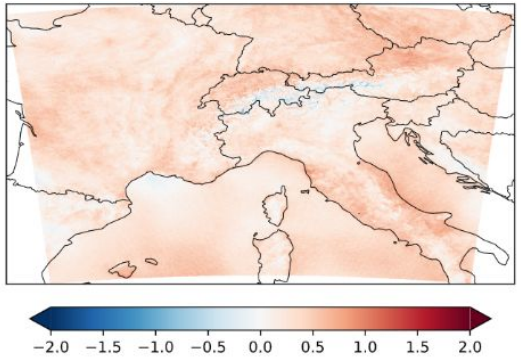
So far, we have reviewed the evaluation frameworks (i.e., what data to use for assessing different aspects of the RCM emulator), but **what metrics should we compute**?

Distributional-based metrics

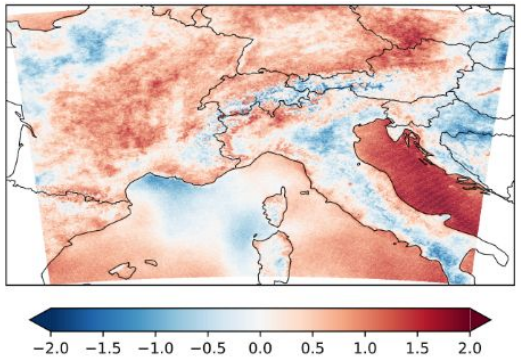
Bias in 2nd Percentile (°C)



Bias in mean (°C)



Bias in 98th Percentile (°C)



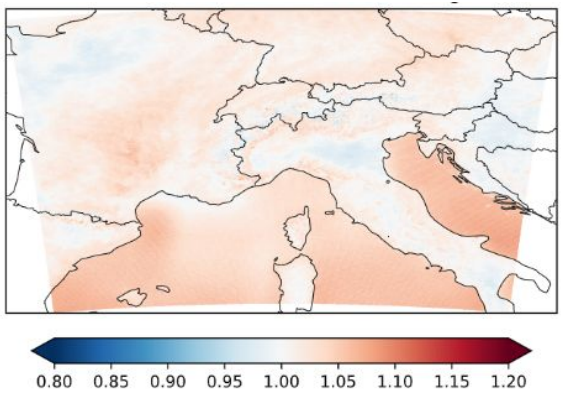
Variable: Temperature
Ground-truth: RCM
Prediction: RCM emulator

Distributional-based Metrics

So far, we have reviewed the evaluation frameworks (i.e., what data to use for assessing different aspects of the RCM emulator), but **what metrics should we compute**?

Distributional-based metrics

Ratio of Standard Deviations



Variable: Temperature
Ground-truth: RCM
Prediction: RCM emulator

Example of Evaluation

Let us suppose that we have trained an RCM emulator on an RCM run over **GCM_1** (**low emissions scenario**)

Variable: Temperature

Example of Evaluation

Let us suppose that we have trained an RCM emulator on an RCM run over **GCM_1** (**low emissions scenario**)

To evaluate the emulator's ability to simulate a **high emissions** scenario, I could proceed as follows (**soft-transferability**)

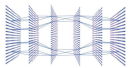
Variable: Temperature

Example of Evaluation

Let us suppose that we have trained an RCM emulator on an RCM run over **GCM_1** (**low emissions scenario**)

To evaluate the emulator's ability to simulate a **high emissions** scenario, I could proceed as follows (**soft-transferability**)

RCM emulator



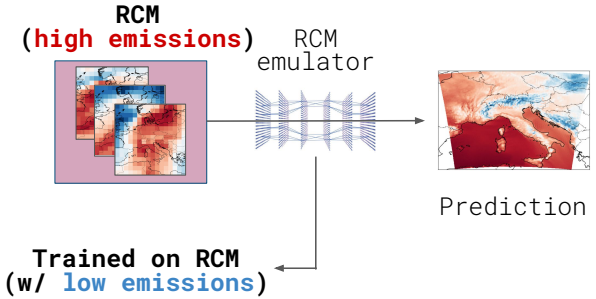
Trained on RCM (w/ low emissions)

Variable: Temperature

Example of Evaluation

Let us suppose that we have trained an RCM emulator on an RCM run over **GCM_1** (**low emissions scenario**)

To evaluate the emulator's ability to simulate a **high emissions** scenario, I could proceed as follows (**soft-transferability**)

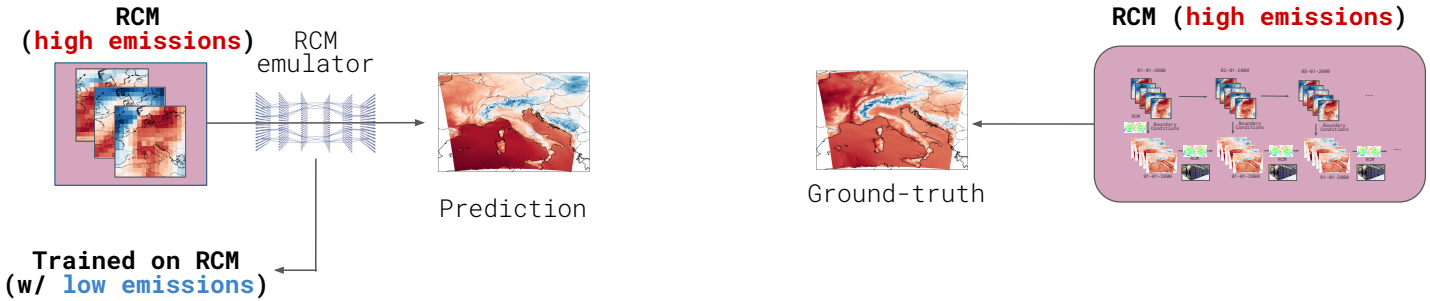


Variable: Temperature

Example of Evaluation

Let us suppose that we have trained an RCM emulator on an RCM run over **GCM_1** (**low emissions scenario**)

To evaluate the emulator's ability to simulate a **high emissions** scenario, I could proceed as follows (**soft-transferability**)

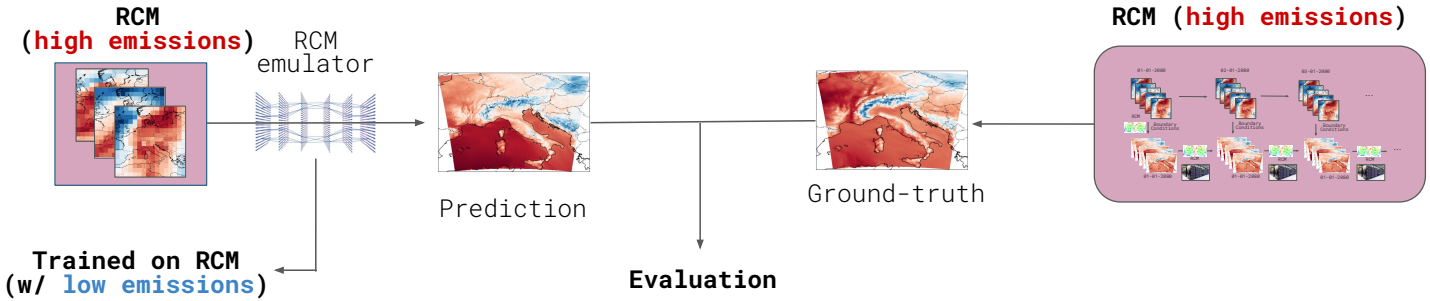


Variable: Temperature

Example of Evaluation

Let us suppose that we have trained an RCM emulator on an RCM run over **GCM_1** (**low emissions scenario**)

To evaluate the emulator's ability to simulate a **high emissions** scenario, I could proceed as follows (**soft-transferability**)

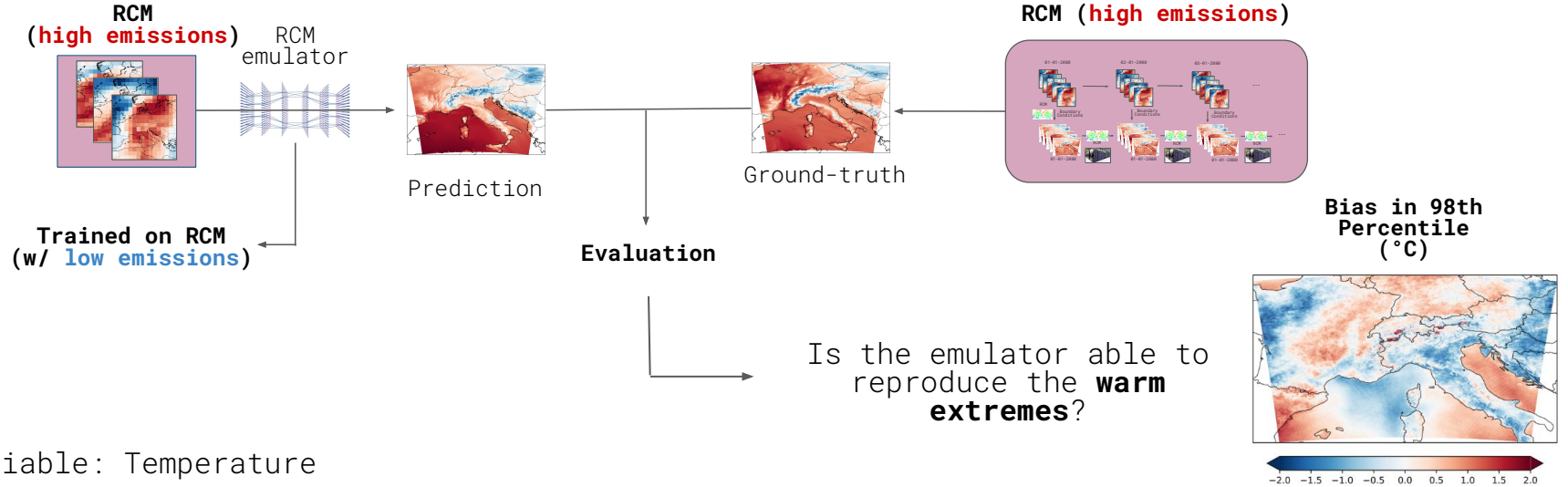


Variable: Temperature

Example of Evaluation

Let us suppose that we have trained an RCM emulator on an RCM run over **GCM_1** (**low emissions scenario**)

To evaluate the emulator's ability to simulate a **high emissions** scenario, I could proceed as follows (**soft-transferability**)



Variable: Temperature

Table of Contents

Benchmark

A benchmark in DL is a **standardized dataset** and **evaluation metric** used to objectively compare and **track the performance of different models**

Benchmark

A benchmark in DL is a **standardized dataset** and **evaluation metric** used to objectively compare and **track the performance of different models**

ImageNet is a large-scale labeled image dataset widely used to train and benchmark deep learning models for **image classification**

Benchmark

A benchmark in DL is a **standardized dataset** and **evaluation metric** used to objectively compare and **track the performance of different models**

ImageNet is a large-scale labeled image dataset widely used to train and benchmark deep learning models for **image classification**

There are multiple benchmarks for **LLMs** that try to **evaluate different dimensions**, such as coding, reasoning, or mathematical skills

LiveBench

A Challenging, Contamination-Free LLM Benchmark

LiveBench will appear as a Spotlight Paper in ICLR 2025.

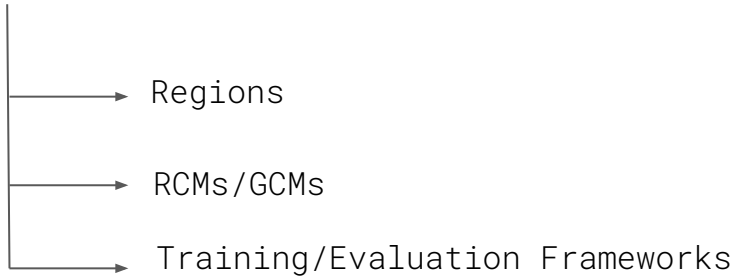
This work is sponsored by Abacus.AI

Leaderboard | Details | Code | Data | Paper

Model	Organization	Global Average	Reasoning Average	Coding Average	Agentic Coding Average
GPT-5 High	OpenAI	78.59	98.17	75.31	43.33
GPT-5 Medium	OpenAI	76.45	96.58	73.25	35.00
GPT-5 Low	OpenAI	75.34	90.47	72.49	41.67
o3 Pro High	OpenAI	74.72	94.67	76.78	31.67
o3 High	OpenAI	74.61	94.67	76.71	36.67
Claude 4.1 Opus Thinking	Anthropic	73.48	93.19	73.96	33.33

Benchmarks in RCM emulation

Despite being a recent field, **various papers** have already been published on **RCM emulation** spanning **different**...

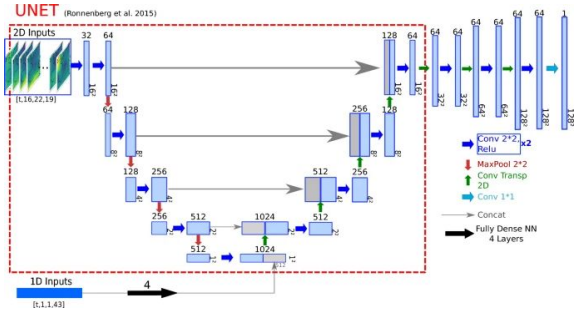


Benchmarks in RCM emulation

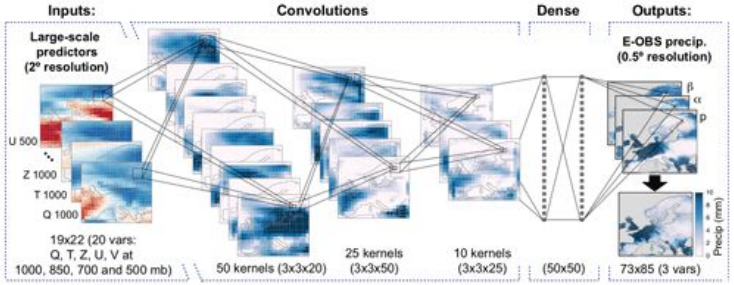
Despite being a recent field, **various papers** have already been published on **RCM emulation** spanning **different**...

- Regions
- RCMs/GCMs
- Training/Evaluation Frameworks

This **complicates the intercomparison of different DL architectures** for RCM emulation



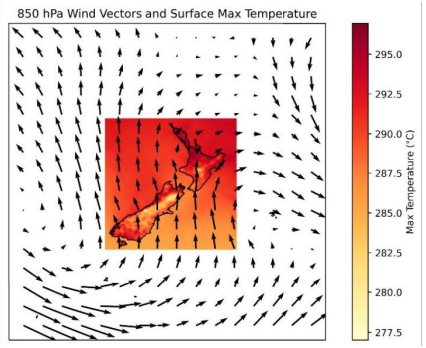
Source: Doury, A., Somot, S., & Gadat, S. (2024). On the suitability of a convolutional neural network based RCM-emulator for fine spatio-temporal precipitation. Climate Dynamics, 62(9), 8587-8613.

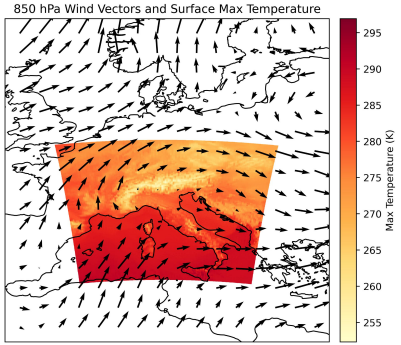


Source: Baño-Medina, J., Manzanas, R., & Gutiérrez, J. M. (2020). Configuration and intercomparison of deep learning neural models for statistical downscaling. Geoscientific Model Development, 13(4), 2109-2124.

What is CORDEX-ML-Bench?

Within CORDEX we are developing **CORDEX-ML-Bench**, a benchmarking dataset for **AI-based regional climate downscaling**





- Different regions
- Different domains
- Different training experiments
- Different evaluation experiments

What is CORDEX-ML-Bench?

In the next session, we will introduce a **first version of this benchmark** (still a **work in progress**) and demonstrate **how to train and evaluate RCM emulators**

What is CORDEX-ML-Bench?

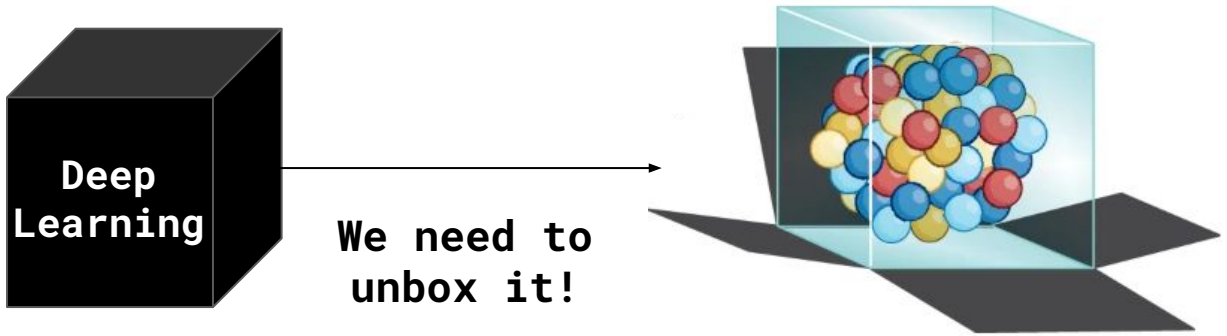
In the next session, we will introduce a **first version of this benchmark** (still a **work in progress**) and demonstrate **how to train and evaluate RCM emulators**

Questions?

Interpretability

Despite being form by simple elements, its **composition** makes deep learning models to be considered **black boxes**

This makes it difficult to gain a **comprehensive** understanding of their inner functioning, **particularly for downscaling climate change projections**



Interpretability

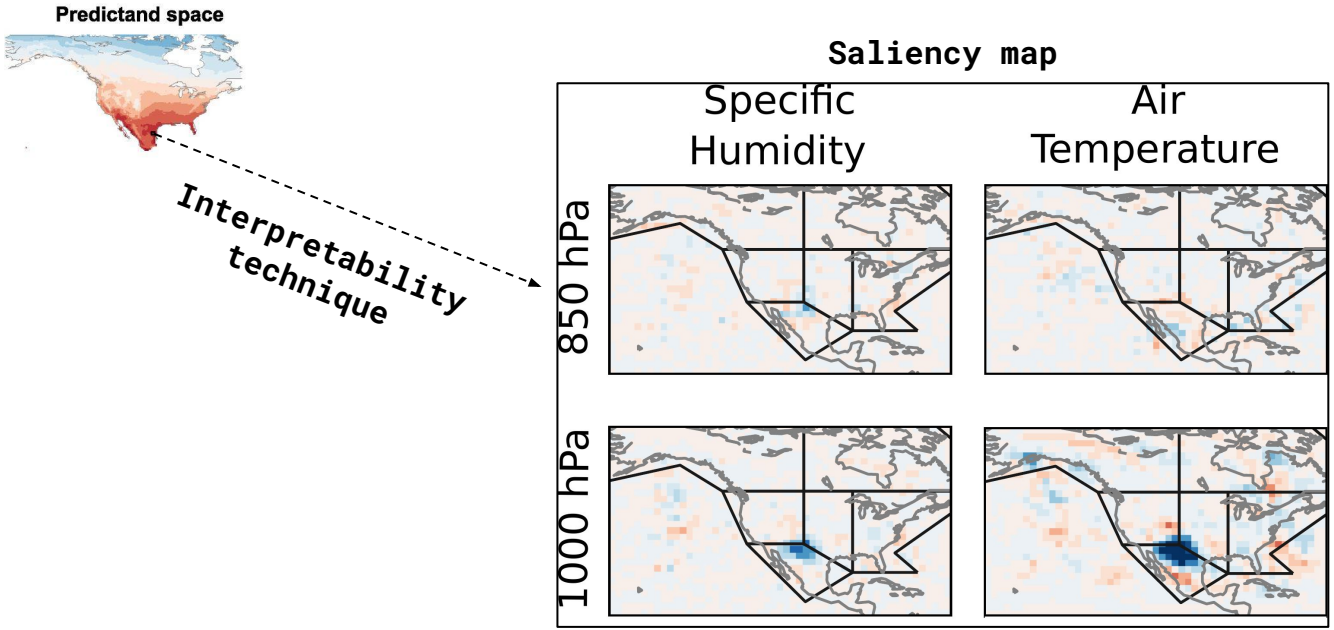
Interpretability techniques emerged in the computer vision field to **explain the functioning and results of deep learning models**

Source: Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016

Saliency maps assign to each feature a **relevance score** representing its influence on the computed prediction

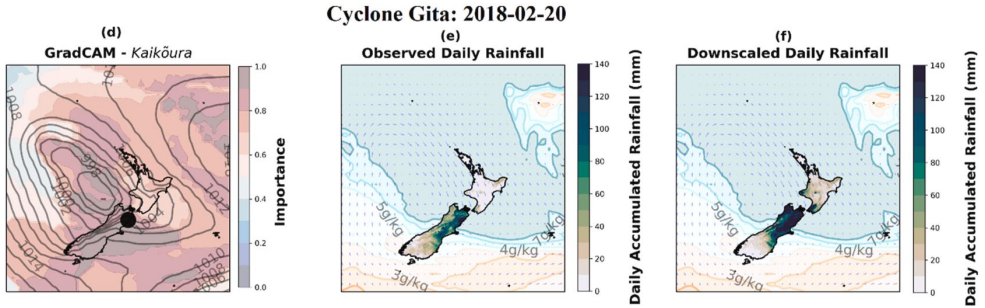
Interpretability

For **statistical downscaling**, **saliency maps** are defined over the **grid-points of the large-scale variables** conforming the **predictor**



Interpretability

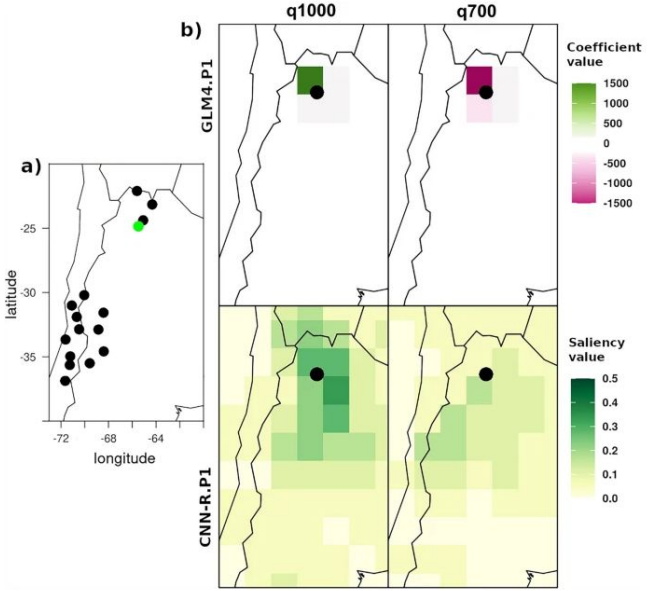
Interpretability techniques have been used to **gain insights into the relationships learnt by the deep learning model**



Rampal et al. (2022)

The deep learning model is taking into account the **cyclone** to downscale the **precipitation extreme**

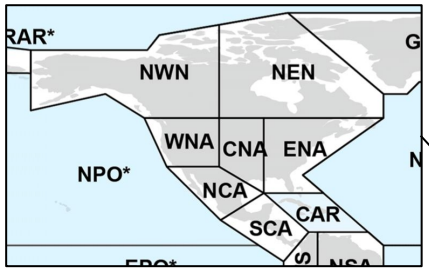
Balmaceda-Huarte et al. (2024)

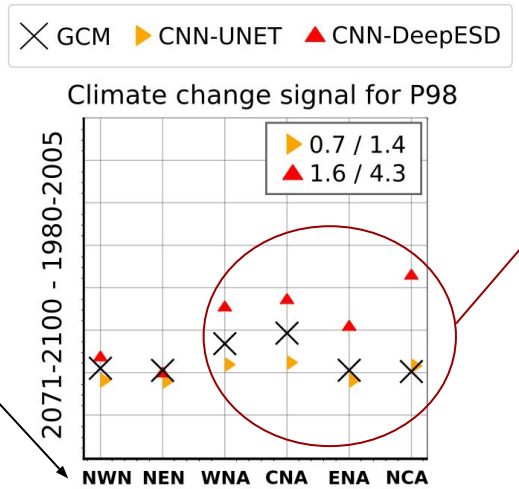


The deep learning model is **learning suitable relationships**

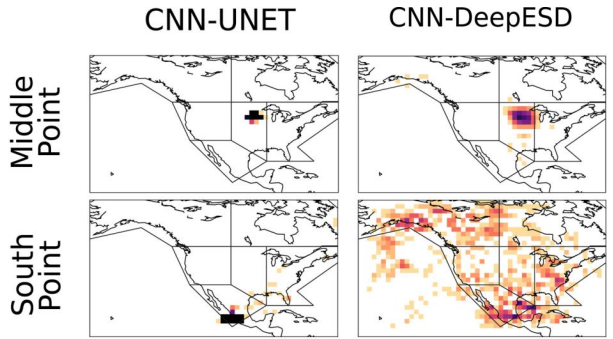
Interpretability

Interpretability techniques have even been used to ***debug*** deep learning models





When downscaling the GCM **(extrapolation)** the **DeepESD overestimates the climate change signal** for the **southern regions** of North America



DeepESD is learning a **spurious pattern for southern regions**, hampering its ability to extrapolate

González-Abad et al. (2023)