Graph neural networks for hourly precipitation projections at the convection permitting scale

Valentina Blasone

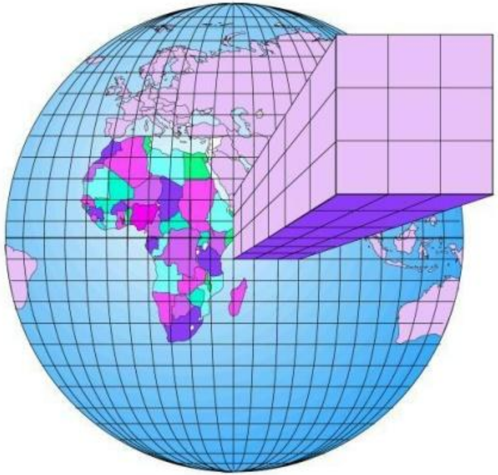
III year PhD Student in ADSAI @ AI Lab, University of Trieste
valentina.blasone@phd.units.it

Supervisor: Luca Bortolussi
Co-supervisors: Erika Coppola, Guido Sanguinetti
Collaborators: Viplove Arora, Serafina Di Gioia

Introduction

Climate Models

Solve the fluid-hydrodynamic equations in all these boxes and exchange information between them



Conservation of momentum, energy, mass and moisture:

$$\frac{\partial \vec{V}}{\partial t} = -(\vec{V} \cdot \nabla)\vec{V} - \frac{1}{\rho}\nabla p - \vec{g} - 2\vec{\Omega} \times \vec{V} + \nabla \cdot (k_\omega \nabla \vec{V}) - \vec{F}_d$$

$$\rho c_p \frac{\partial T}{\partial t} = -\rho c_p (\vec{V} \cdot \nabla) T - \nabla \cdot \vec{R} + \nabla \cdot (k_\tau \nabla T) + C + S$$

$$\frac{\partial \rho}{\partial t} = -(\vec{V} \cdot \nabla)\rho - \rho(\nabla \cdot \vec{V})$$

$$\frac{\partial q}{\partial t} = -(\vec{V} \cdot \nabla) q + \nabla \cdot (k_q \nabla q) + S_q + E$$

Equation of state:

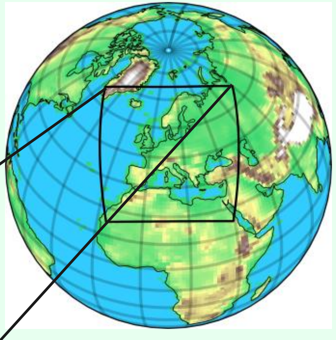
$$p = \rho R_d T$$

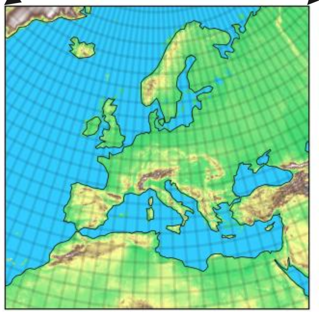
$V = velocity$
$T = temperature$
$p = pressure$
$\rho = density$
$q = specific\ humidity$
$g = gravity$
$\Omega = rotation\ of\ Earth$
$F_d = drag\ force\ of\ Earth$
$R = radiation\ vector$
$C = conductive\ heating$
$c_p = heat\ capacity, constant\ p$
$E = evaporation$
$S = latent\ heating$
$S_q = phase\ change\ source$
$k = diffusion\ coefficients$
$R_d = dry\ air\ gas\ constant$

GCMs and RCMs

Global Climate Models (GCMs)

Low resolution 50-250 km
Global to sub-continental scale, too coarse for local impacts of global climate change

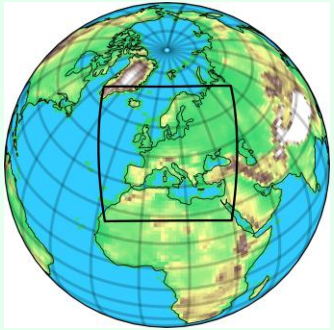


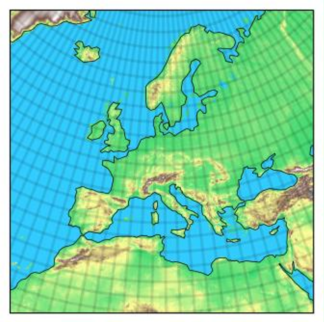


Regional Climate Models (RCMs)

High resolution 50-1km
Regional scale, driven by a GCM simulation at the domain borders

CPMs

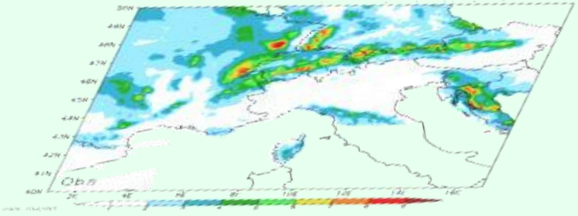




GCM
or RCM

Convection Permitting Models (CPMs)

Very high resolution ≤ 3km



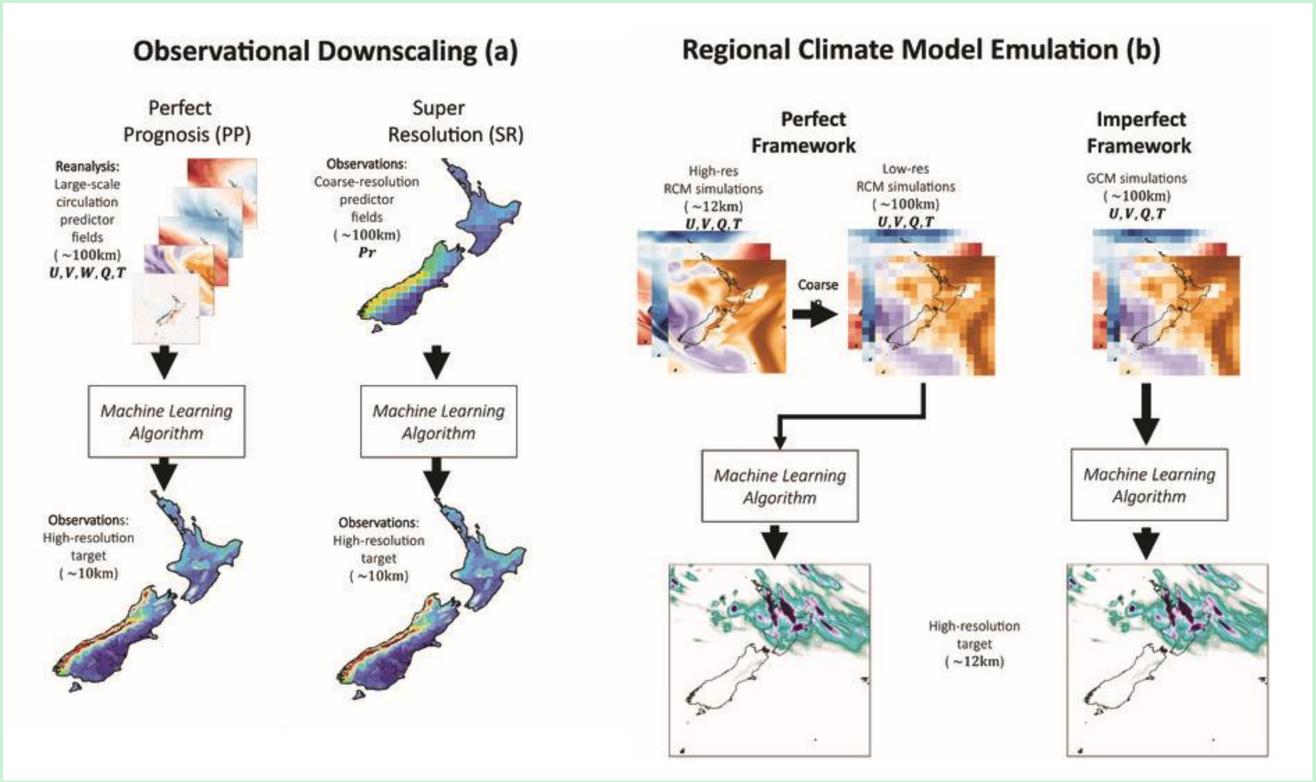
Why deep learning?

Often **long climate projections** are needed, or **many simulations** are required to estimate the climate projections uncertainty

High resolution → **High computational cost**

Deep learning exploits the available data and can be used for <u>observational downscaling</u> and <u>regional climate model emulation</u> in a **computationally efficient** way

Downscaling and emulation

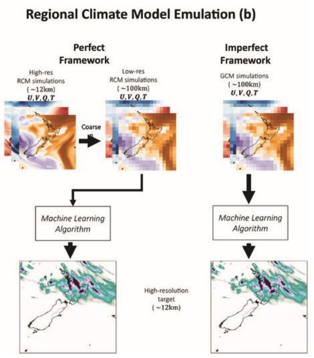


Rampal et al., 2024

Emulation: existing approaches

All emulators are trained using data from climate models:

- **Predictors**: upscaled large-scale variables from the same RCM, or driving GCM large-scale fields
- **Target**: RCM variables (temperature, precipitation, ...)



Advantages: same type of data in learning and inference

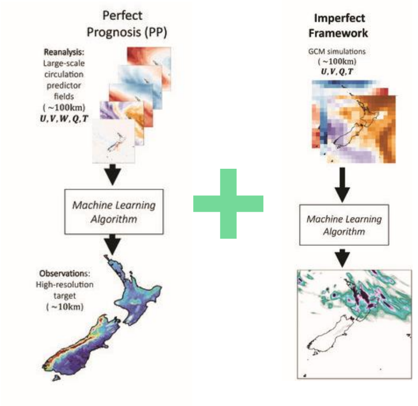
Disadvantages: learn the emulator of a specific climate model, incorporates bias of the climate model

Emulation: our approach

Train the emulator for observational downscaling with perfect prognosis (PP):

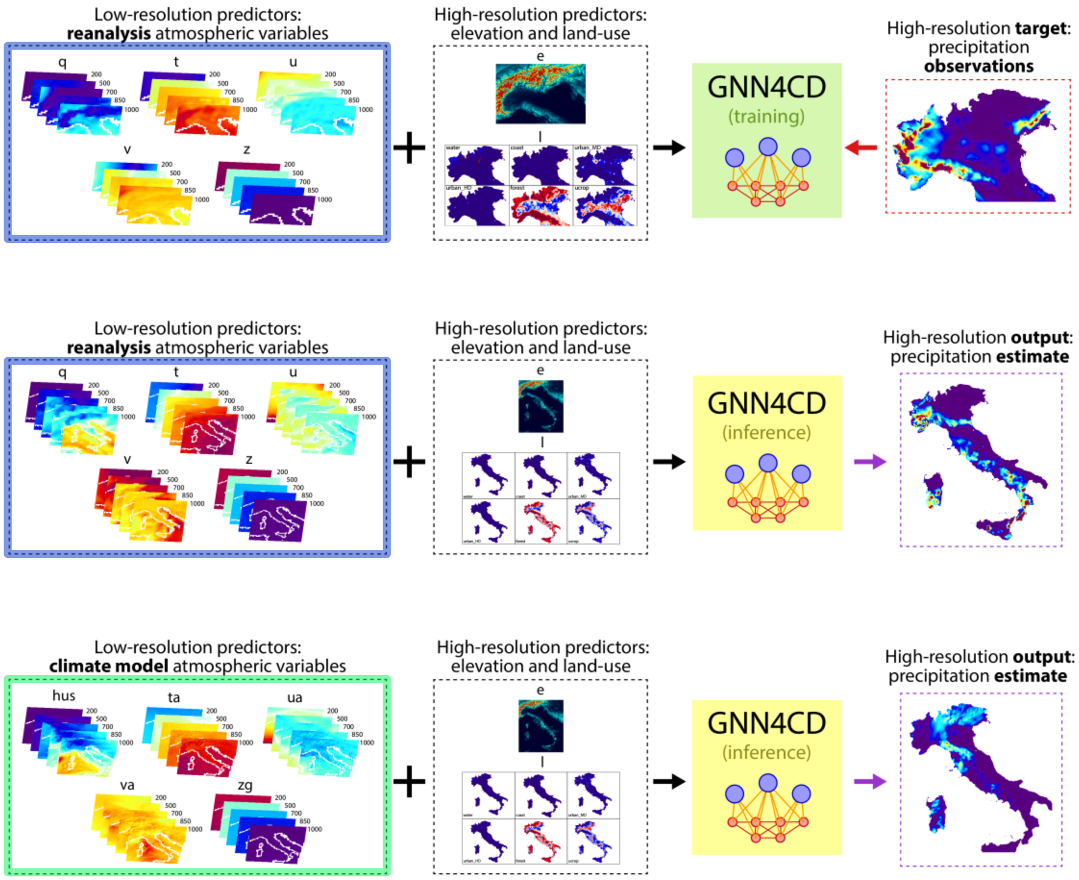
- **Predictors**: driving reanalysis large-scale
- **Target**: high-res observed variable

Then, use RCM input data only during inference



Advantages: avoid model-specificity and bias in training

Disadvantages: more difficult problem, different type of data in learning and inference



Training: observational downscaling

Inference: observational downscaling

Inference: emulation

* **GNN4CD**: Graph Neural Networks for Climate Downscaling

Precipitation observational downscaling

Precipitation is challenging

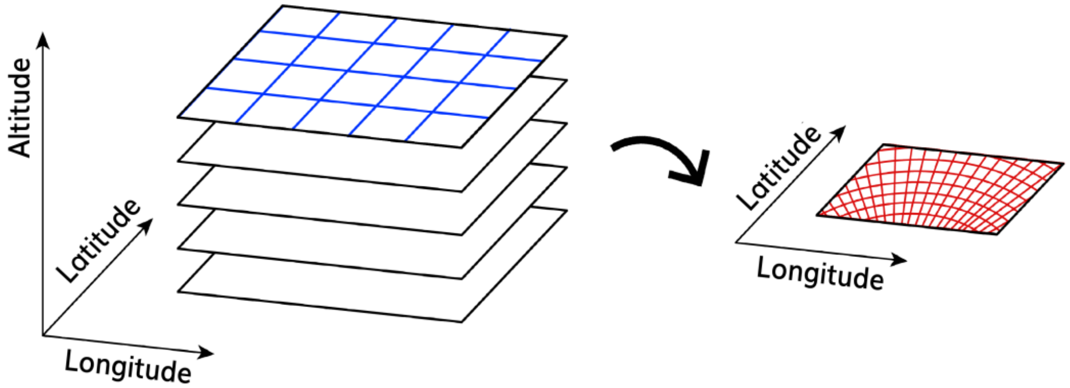
Severe precipitation is a complex phenomenon, related to convective systems with complex and non-linear airflow motion

High resolution is crucial to capture convection phenomena and correctly quantify severe precipitation and extreme events

Real world observational datasets are rare and often need careful preprocessing (missing data, …)

The main task

~25 km
1 hour



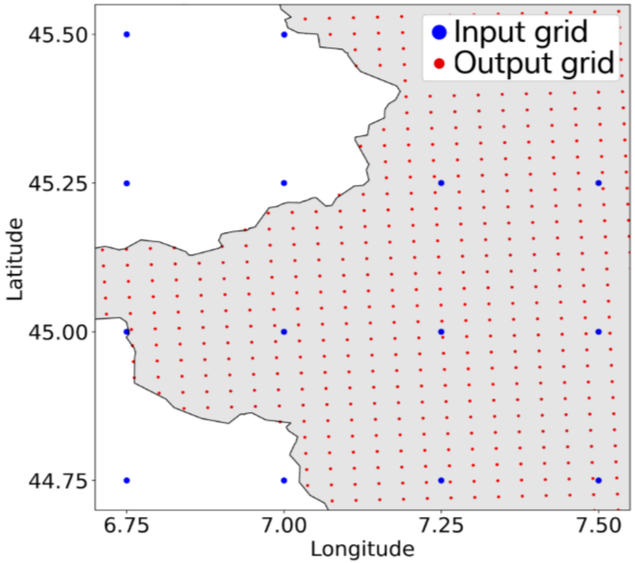
3 km
1 hour

LOW-RESOLUTION
ATMOSPHERIC DATA

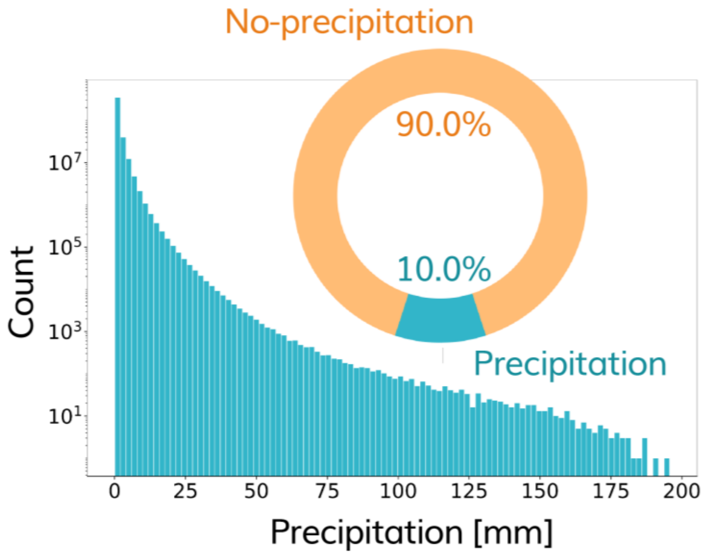
HIGH-RESOLUTION
PRECIPITATION

Main research questions

How to deal with the different resolutions?



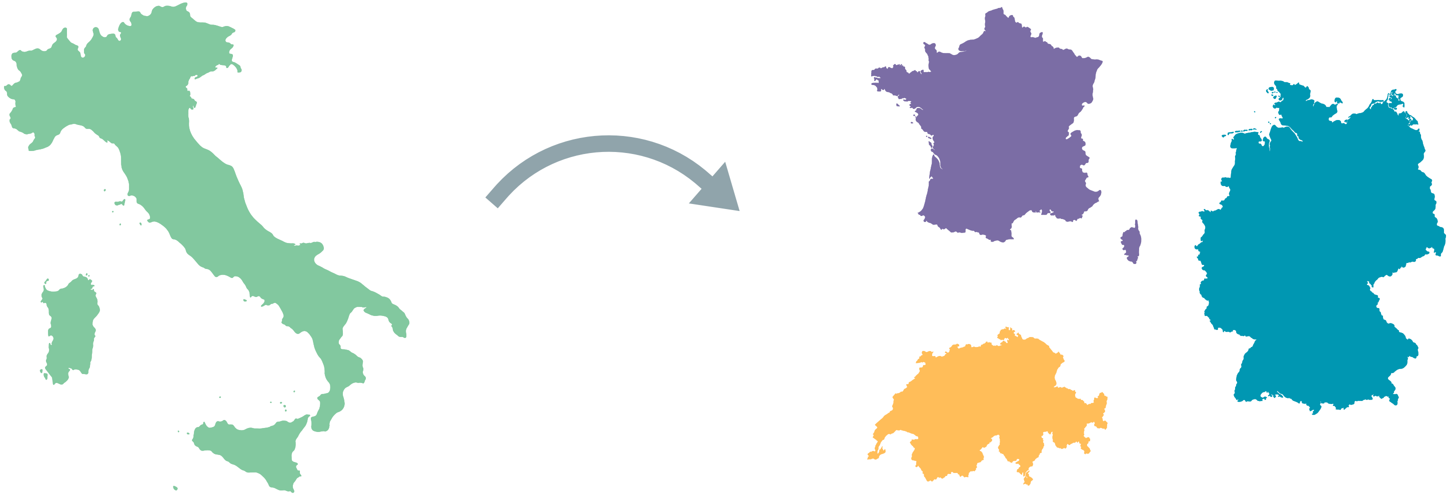
How to deal with imbalanced and skewed data?



Main research questions

Transferability

to other domains?



Which data

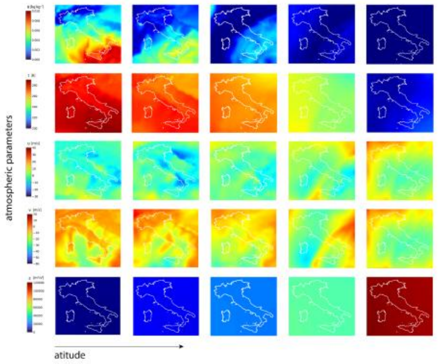
	Variable	Symbol	Unit	Pressure Levels [hPa]	Space	Time
P	Specific humidity	q, hus	[kg kg-1]	1000; 850; 700; 500; 200	0.25°	1hr
	Temperature	t, ta	[K]	1000; 850; 700; 500; 200	0.25°	1hr
	Eastward wind	u, ua	[m/s]	1000; 850; 700; 500; 200	0.25°	1hr
	Northward wind	v, va	[m/s]	1000; 850; 700; 500; 200	0.25°	1hr
	Geopotential	z, zg	[m^2/s^2]	1000; 850; 700; 500; 200	0.25°	1hr
	Elevation	e	[m]	Surface	3km	-
	Land-use	l	[%]	Surface	3km	-
T	Precipitation	pr	[mm]	Surface	3km	1hr

Which data

Input datasets

Target dataset

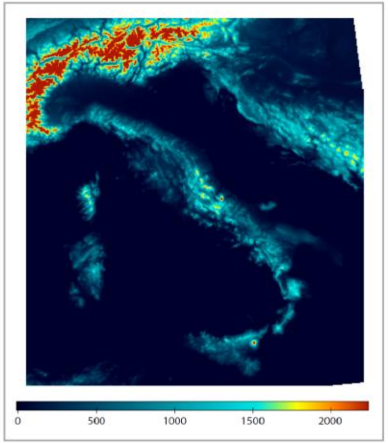
ERA5 REANALYSIS
(~25 km)



HUMIDITY, TEMPERATURE, WIND, GEOPOTENTIAL

4D: lon, lat, altitude, time (hourly)

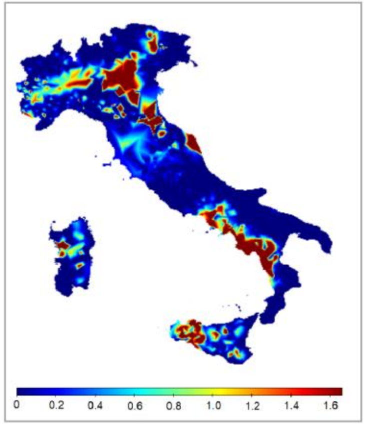
TOPOGRAPHIC ELEVATION
(3 km)



+ LAND USE

2D: lon, lat

GRIPHO OBSERVATIONS
(3 km)



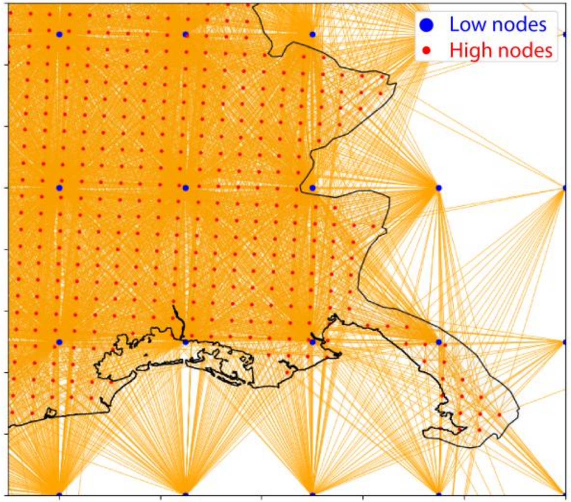
PRECIPITATION

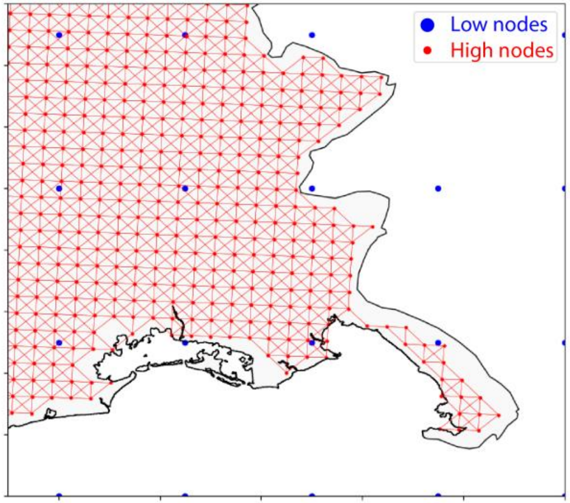
3D: lon, lat, time (hourly)

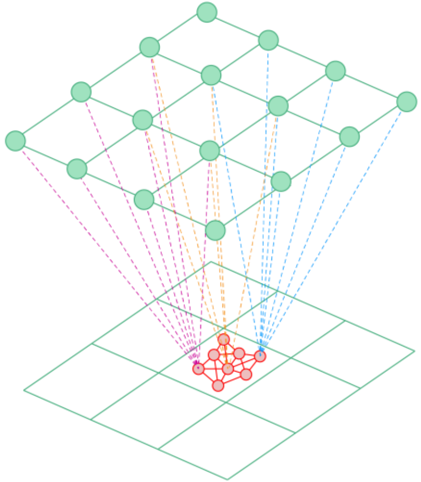
The GNN4CD model

Graph conceptualization

● Low nodes (~25 km)
● High nodes (3 km)
→ Low-to-High edges
↔ High-within-High edges



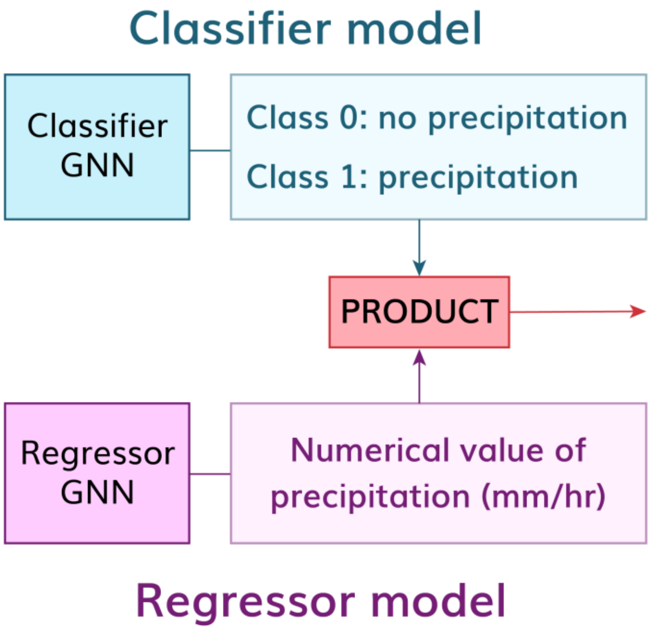




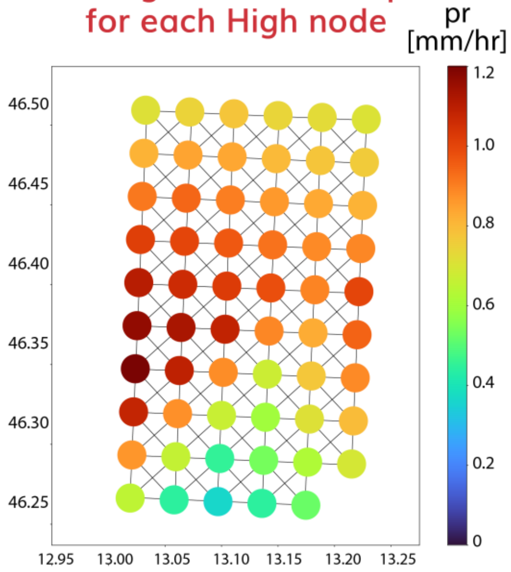
GNN4CD-RC

Trained on all data

Trained on pr ≥ 0.1mm



Precipitation prediction for a single 1 hr timestep * for each High node



* Predictors at time $[t_{i-24}, \ldots, t_i]$ are used to derive the estimate at time t_i

GNN4CD-R-all

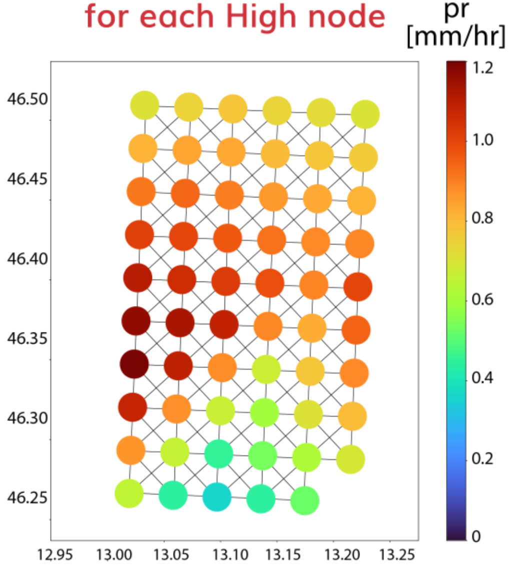
Trained on all data

Regressor GNN

Numerical value of precipitation (mm/hr)

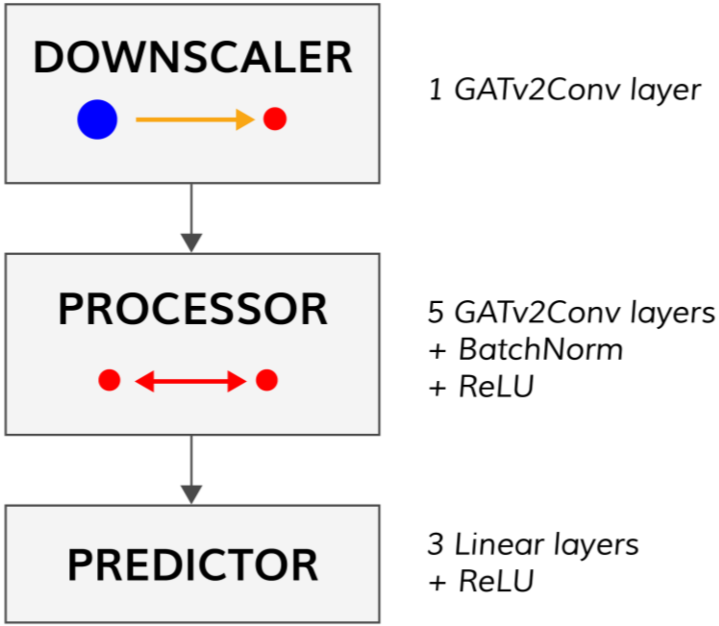
Regressor model

Precipitation prediction for a single 1 hr timestep * for each High node



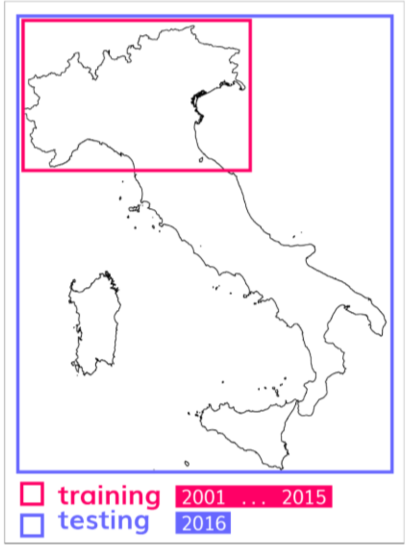
* Predictors at time $[t_{i-24}, \ldots, t_i]$ are used to derive the estimate at time t_i

Architecture



- The model is implemented using **pytorch** and **pytorch geometric**
- This model is the synthesis of multiple experiments

Training/testing



~400 Low nodes
~14000 High nodes

~1000 Low nodes
~33000 High nodes

- **MSE + α QMSE loss** (regressor)
- **Focal loss** (classifier)

- **Moderately long training**: 50 epochs ~24h using 4GPUs on Leonardo

- **Fast inference**: precipitation estimates for one year takes just a few minutes

Losses formulation

Quantised MSE Loss

$$\text{QMSE} = \sum_{j}^{B} \frac{1}{|\Omega_j|} \sum_{i \in \Omega_j} (y_i - \widehat{y_i})^2$$

B: number of bins (bins are defined over the training data domain); j: bin index, from 1 to B

Ω_j: set of target indices whose values fall within bin j (defined dynamically over the batches)

y_i: predicted value; $\hat{y}_i$: ground-truth target value

Focal Loss

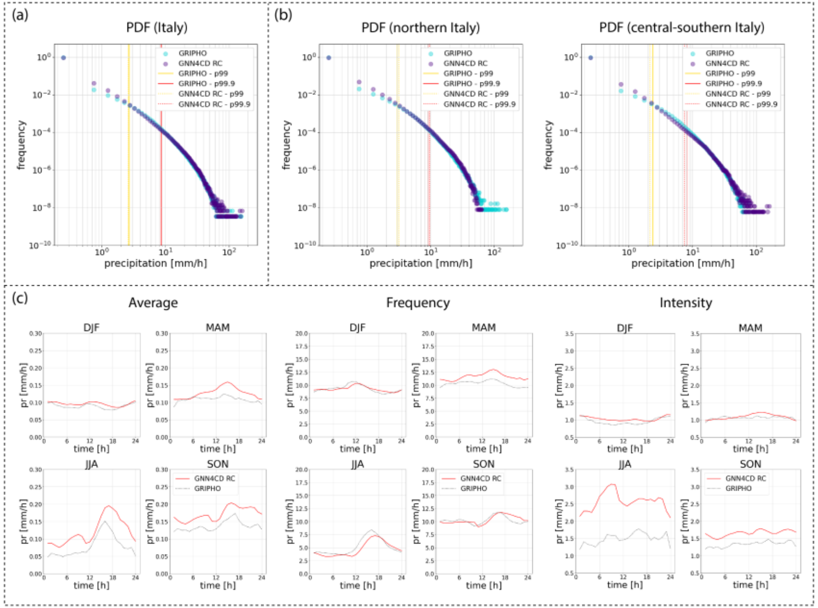
$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \cdot log(p_t)$$

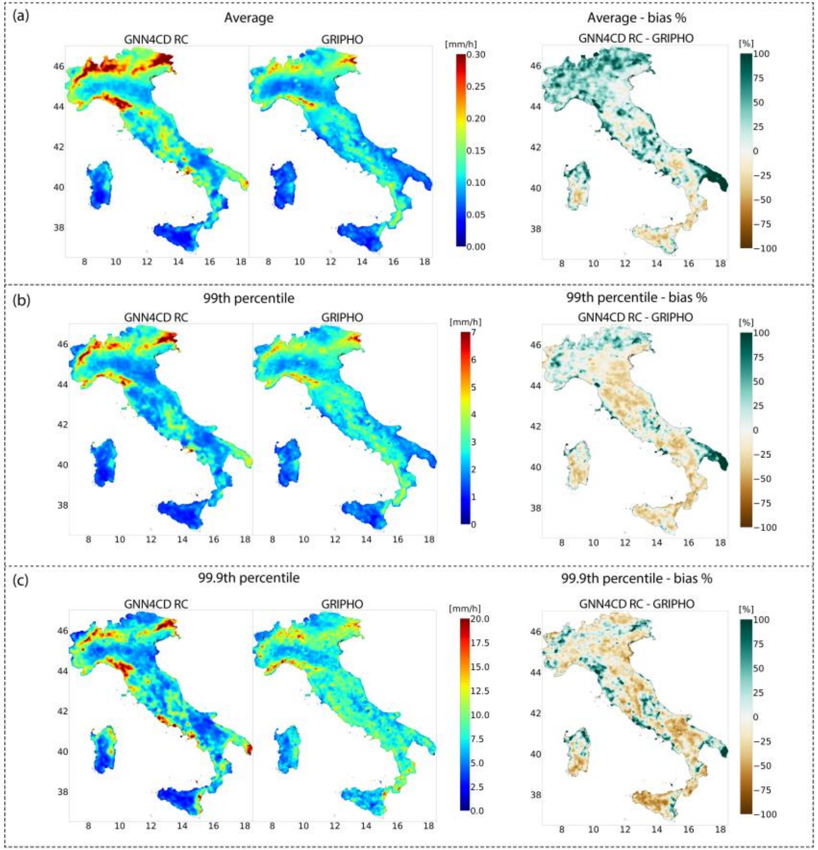
$$p_t = \begin{cases} p & if\ y = 1 \\ 1 - p & otherwise \end{cases}$$

$y \in \{0,1\}$: the ground-truth class; $p \in [0,1]$: the model estimated probability for the class with label $y = 1$

GNN4CD for observational downscaling

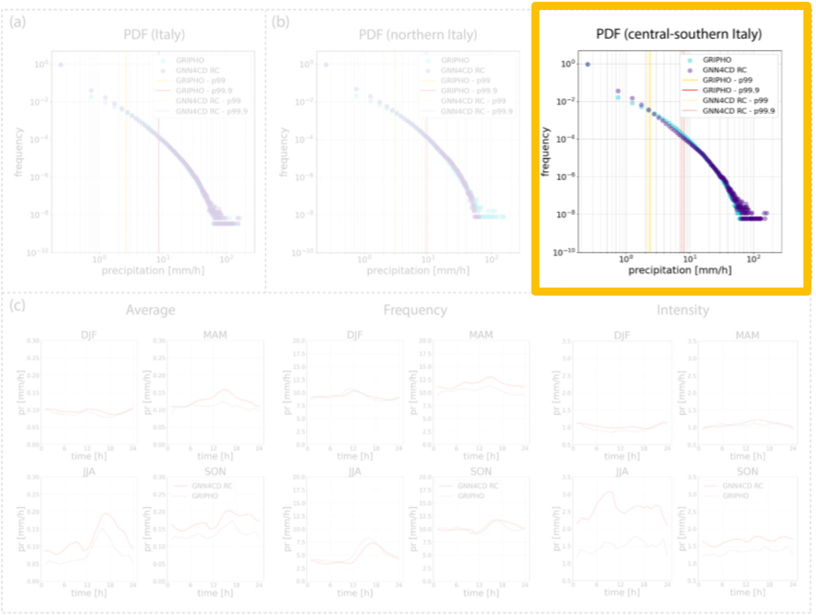
Observational downscaling: RC

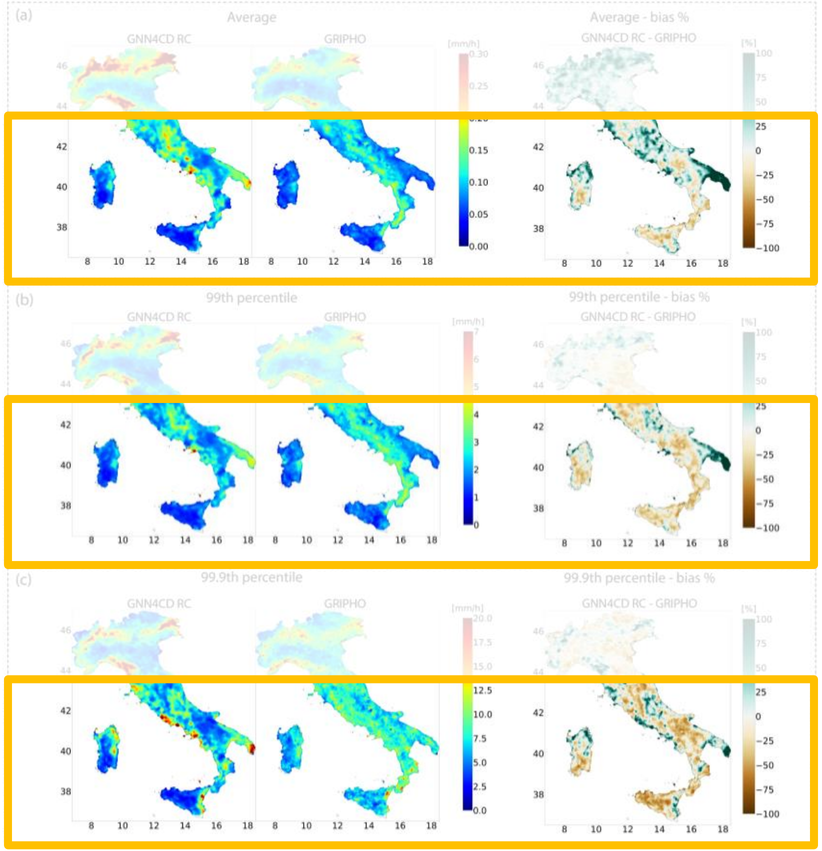




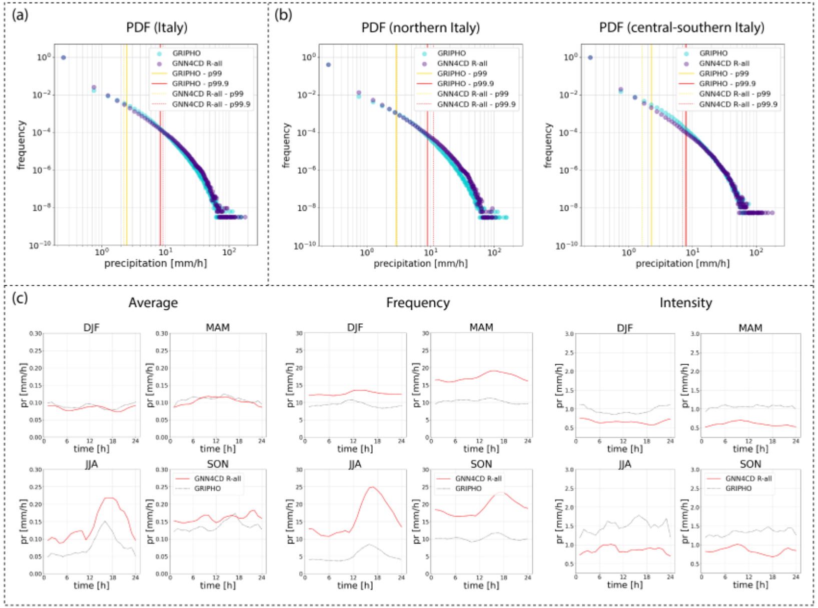
Observational downscaling: RC

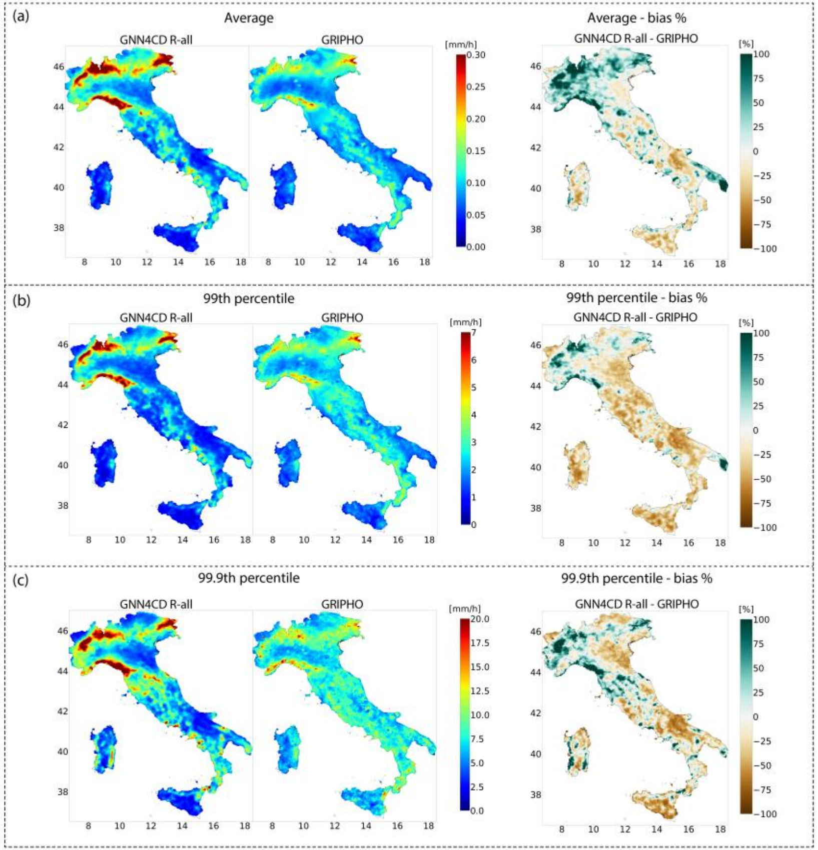
Spatial transferability





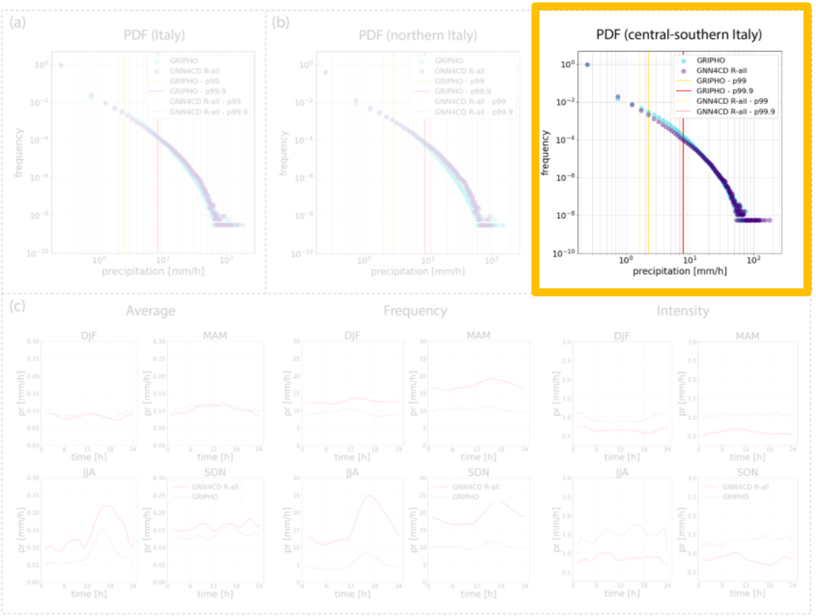
Observational downscaling: R-all

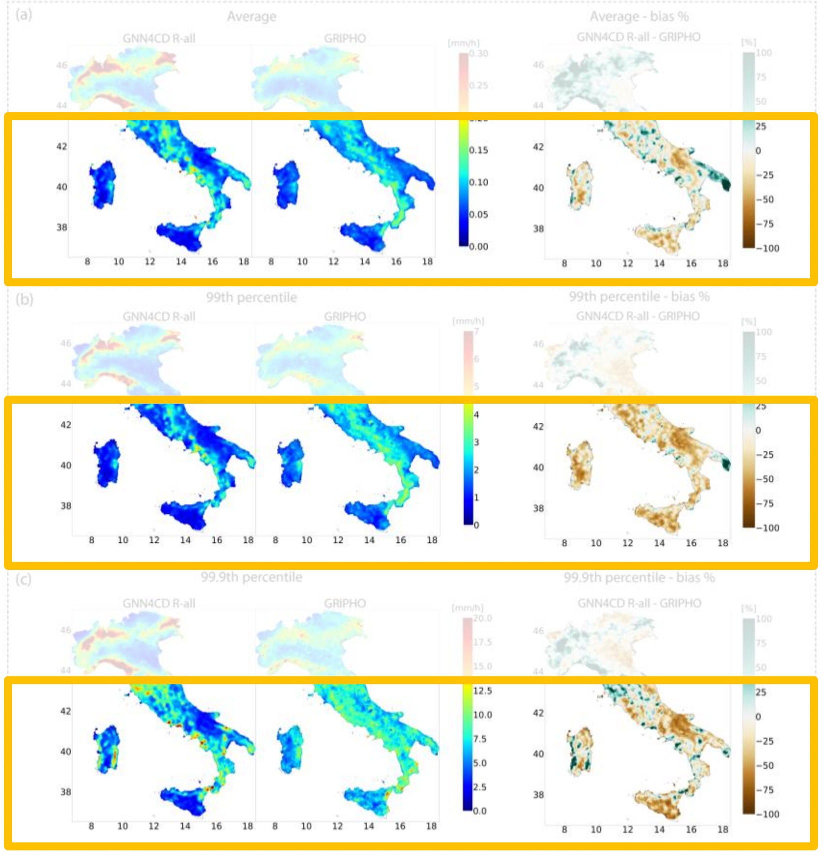




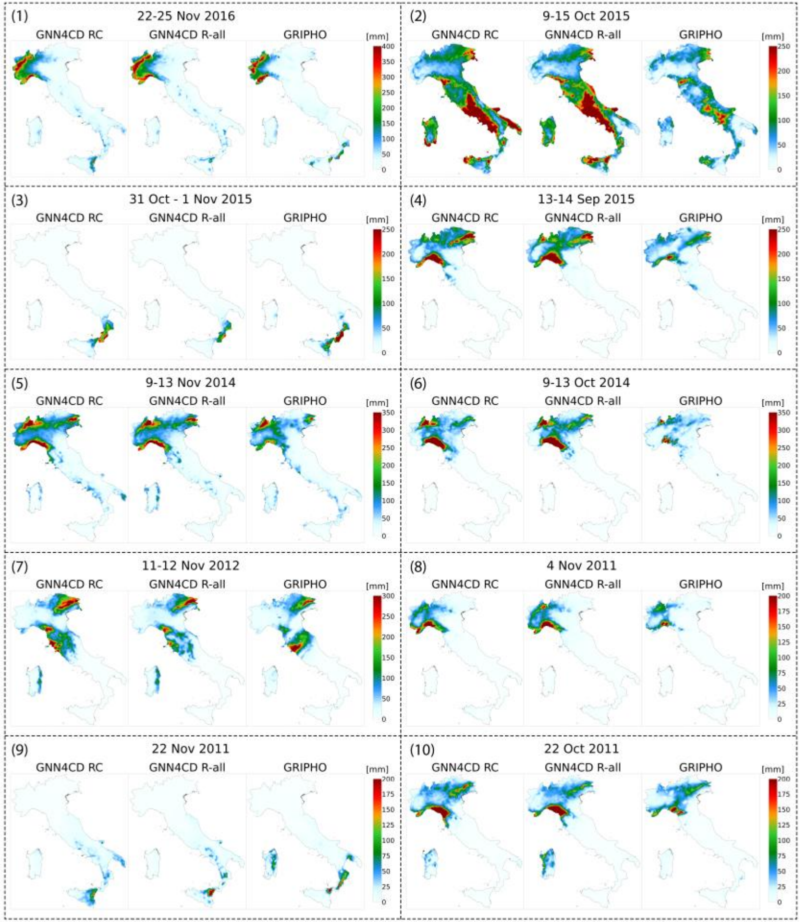
Observational downscaling: R-all

Spatial transferability

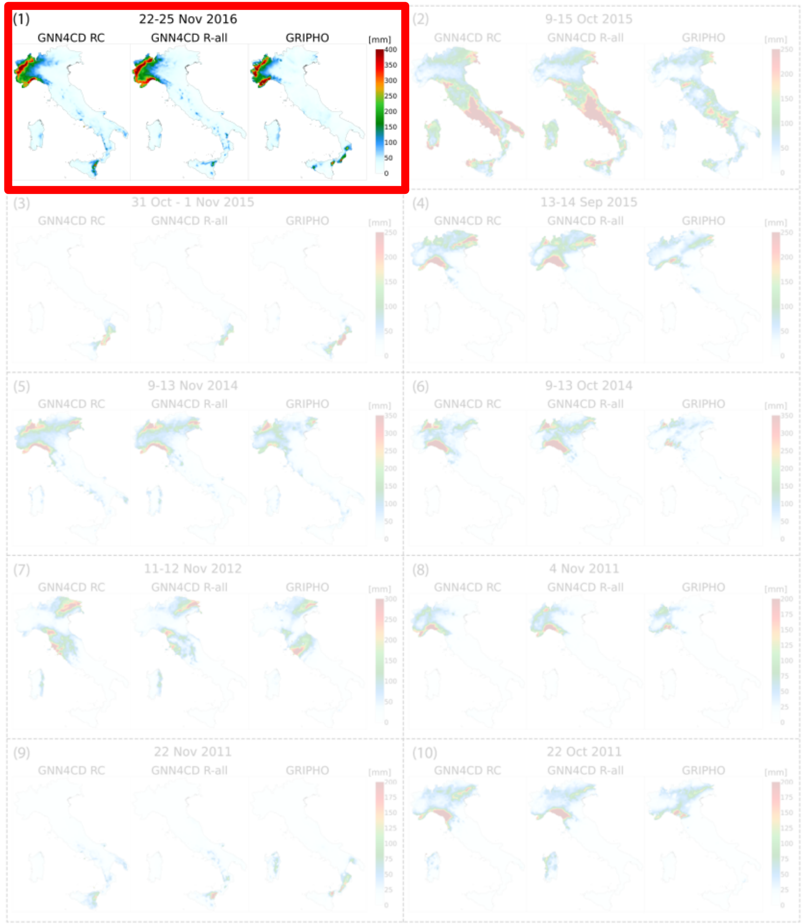




Floods: RC and R-all



Floods: RC and R-all

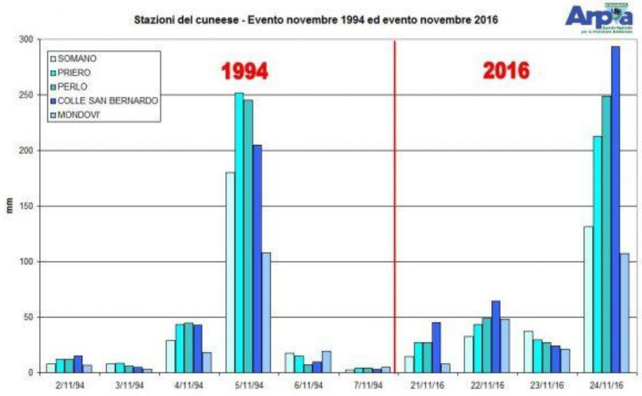


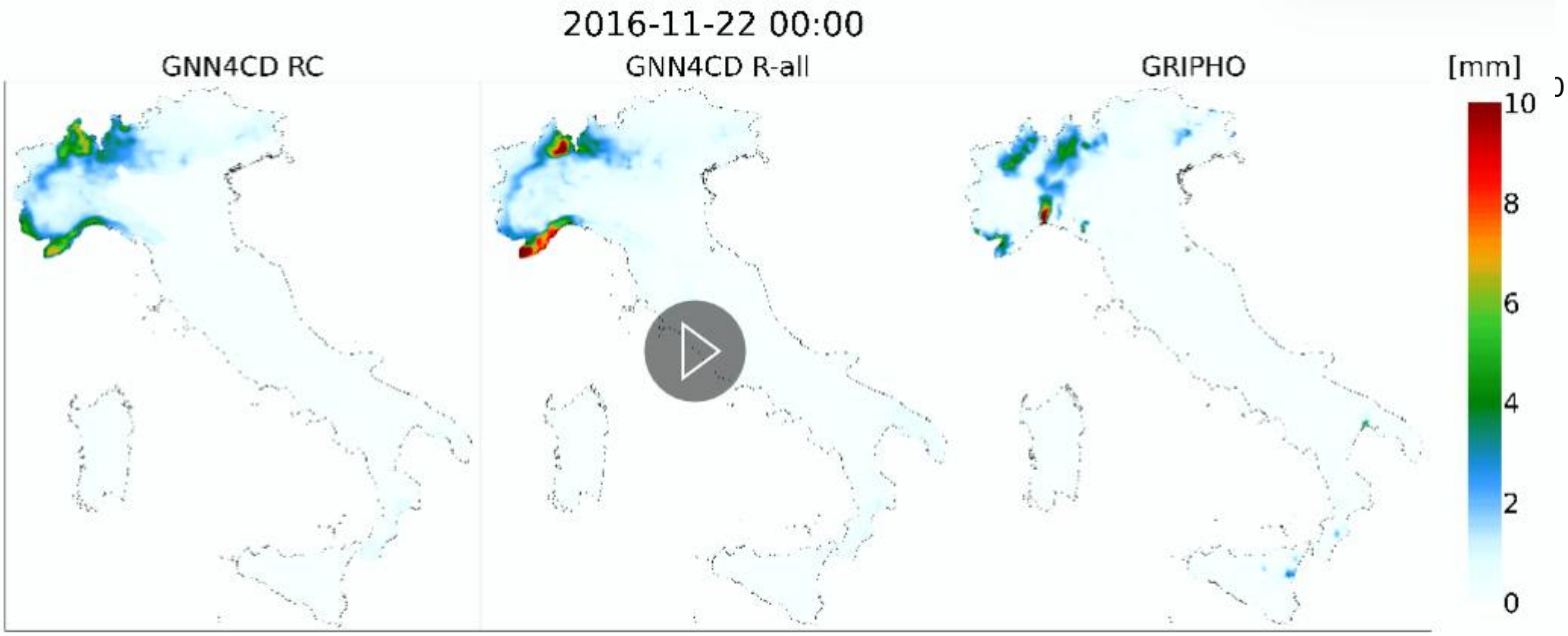
22-25 Nov 2016

Northern Italy experienced **severe flooding** due to prolonged and intense rainfall, particularly affecting the regions of **Piedmont** and **Liguria**.

On 25 November 2016 the **Po** and **Tanaro** rivers **flooded** the surrounding areas, forcing 400 people to be evacuated.

In Liguria both **floods** and **landslides** were reported.

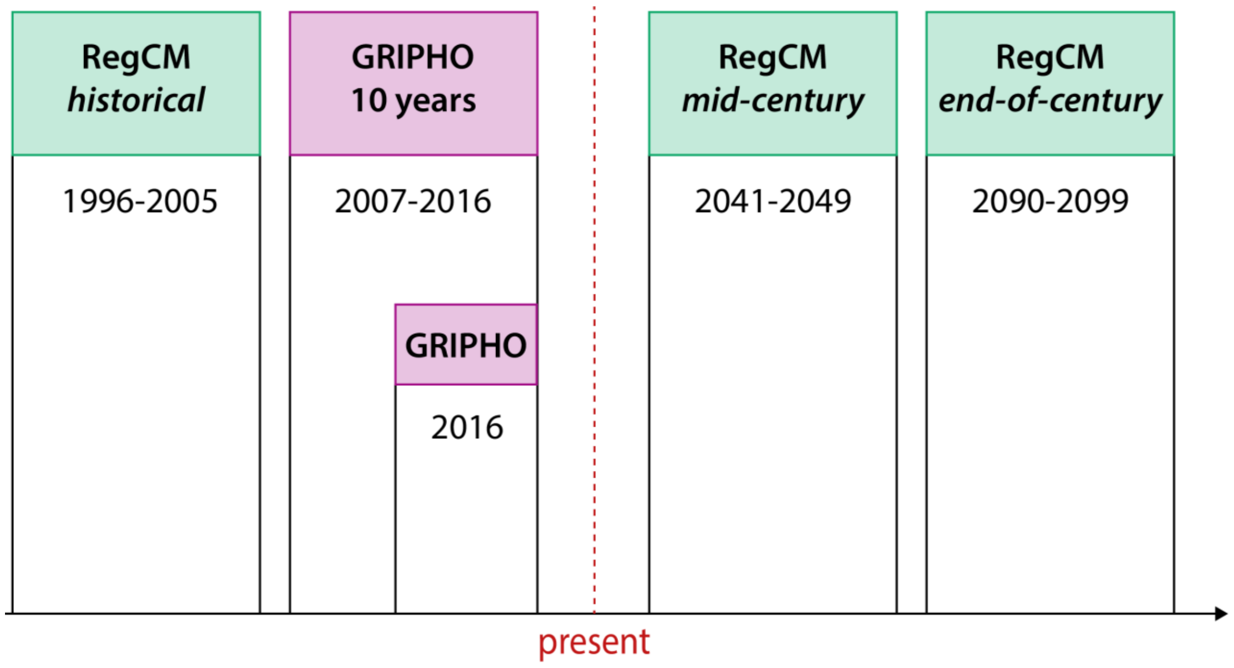



2016-11-22 00:00
GNN4CD RC
GNN4CD R-all
GRIPHO
[mm]
10
8
6
4
2
0

GNN4CD for emulation

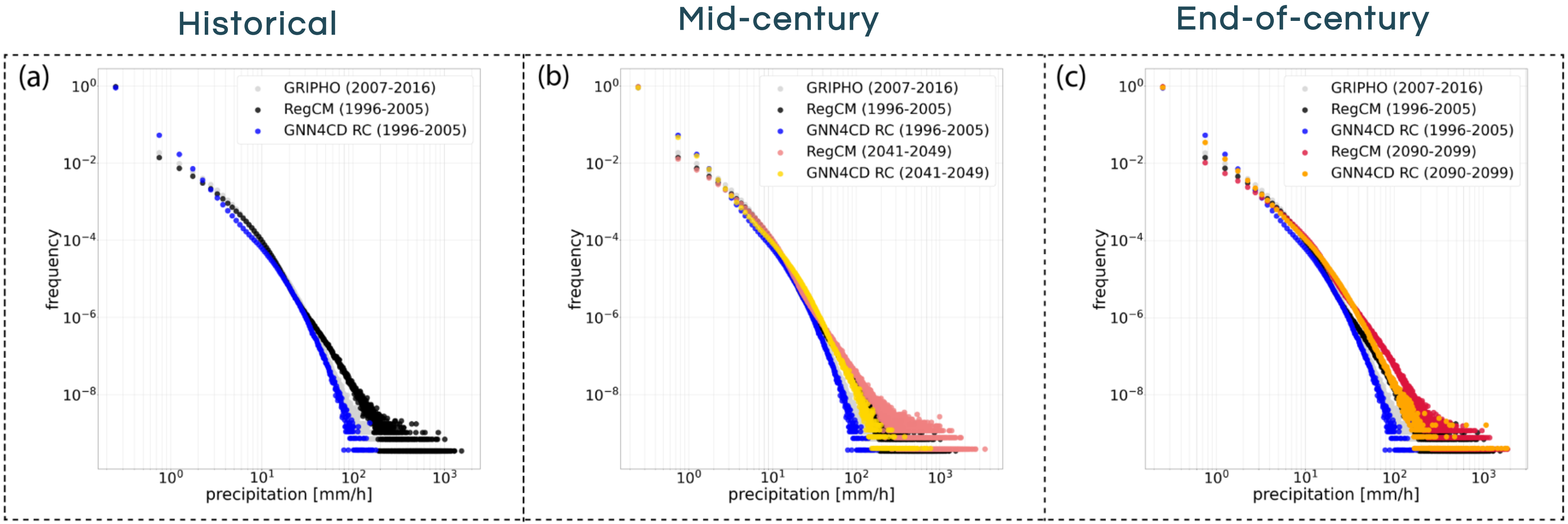
Emulation

Predictors are climate model data



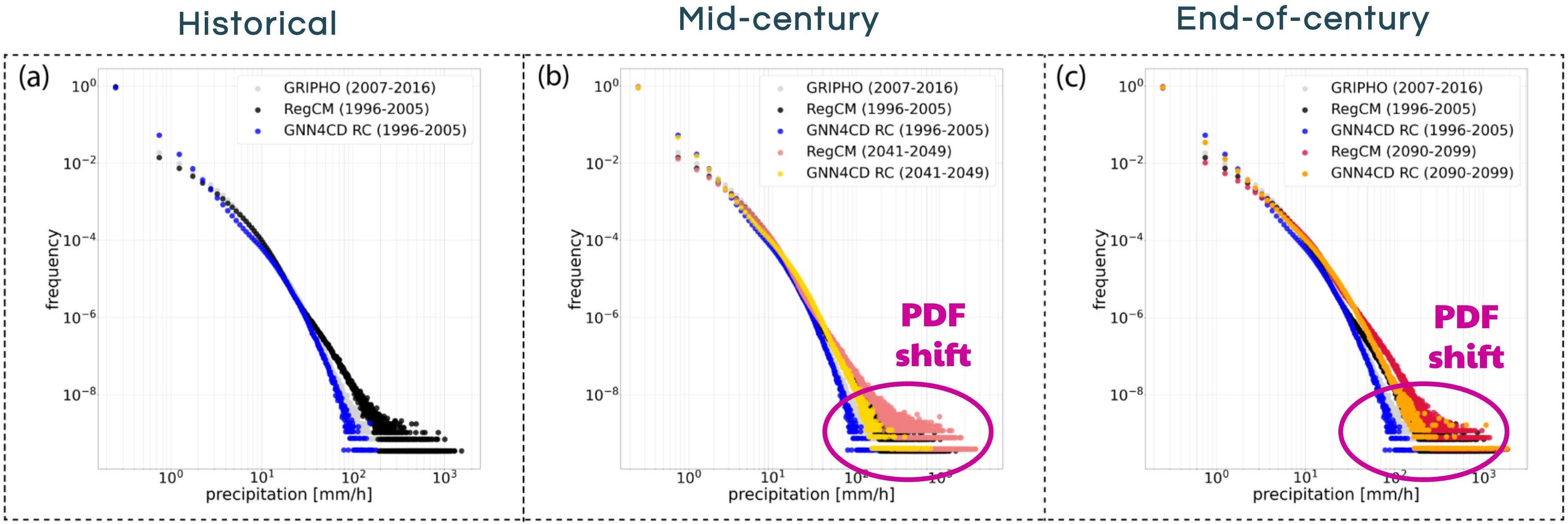
Emulation: RC

Precipitation distribution



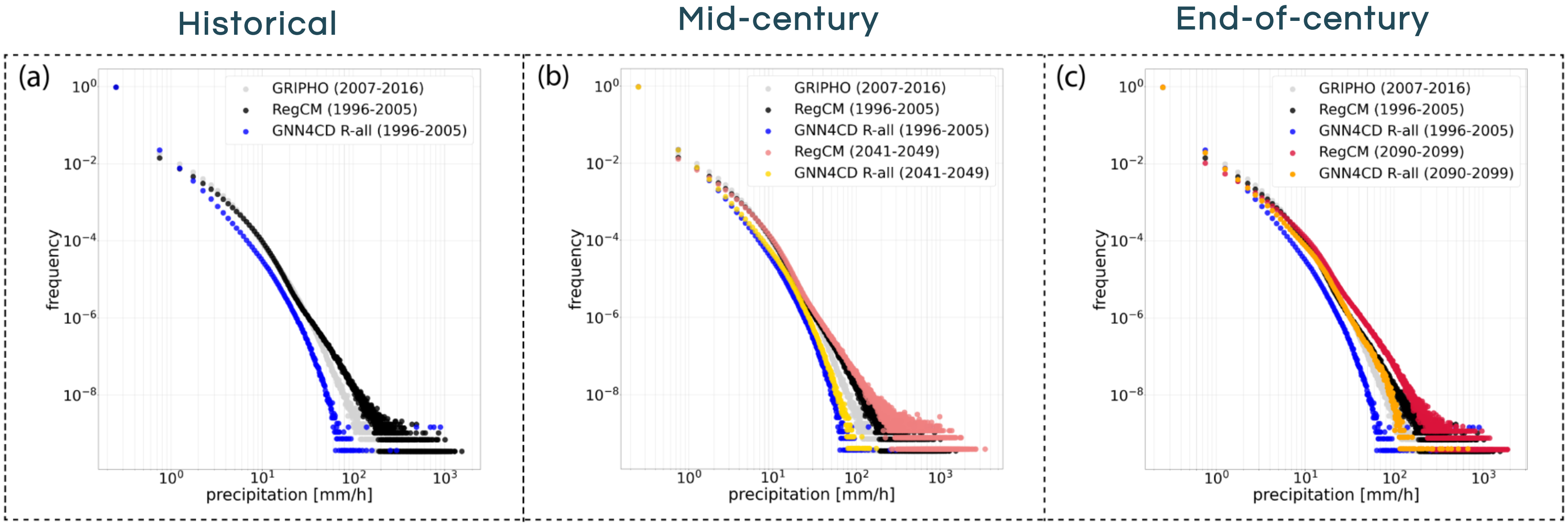
Emulation: RC

Precipitation distribution



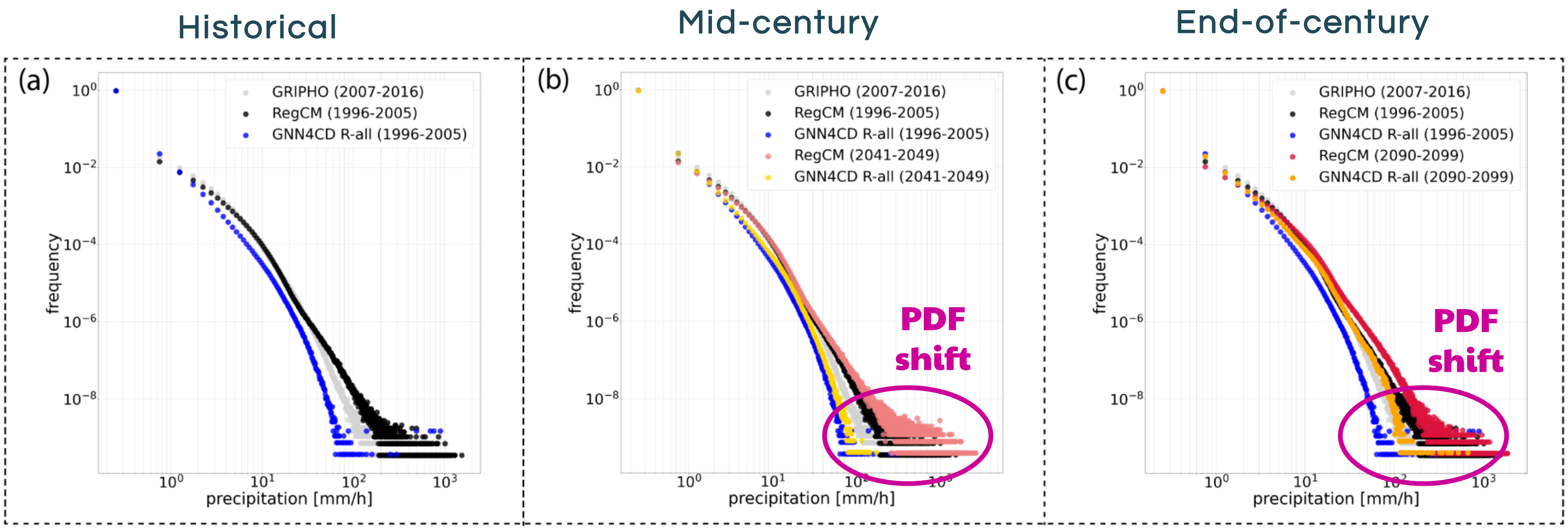
Emulation: R-all

Precipitation distribution



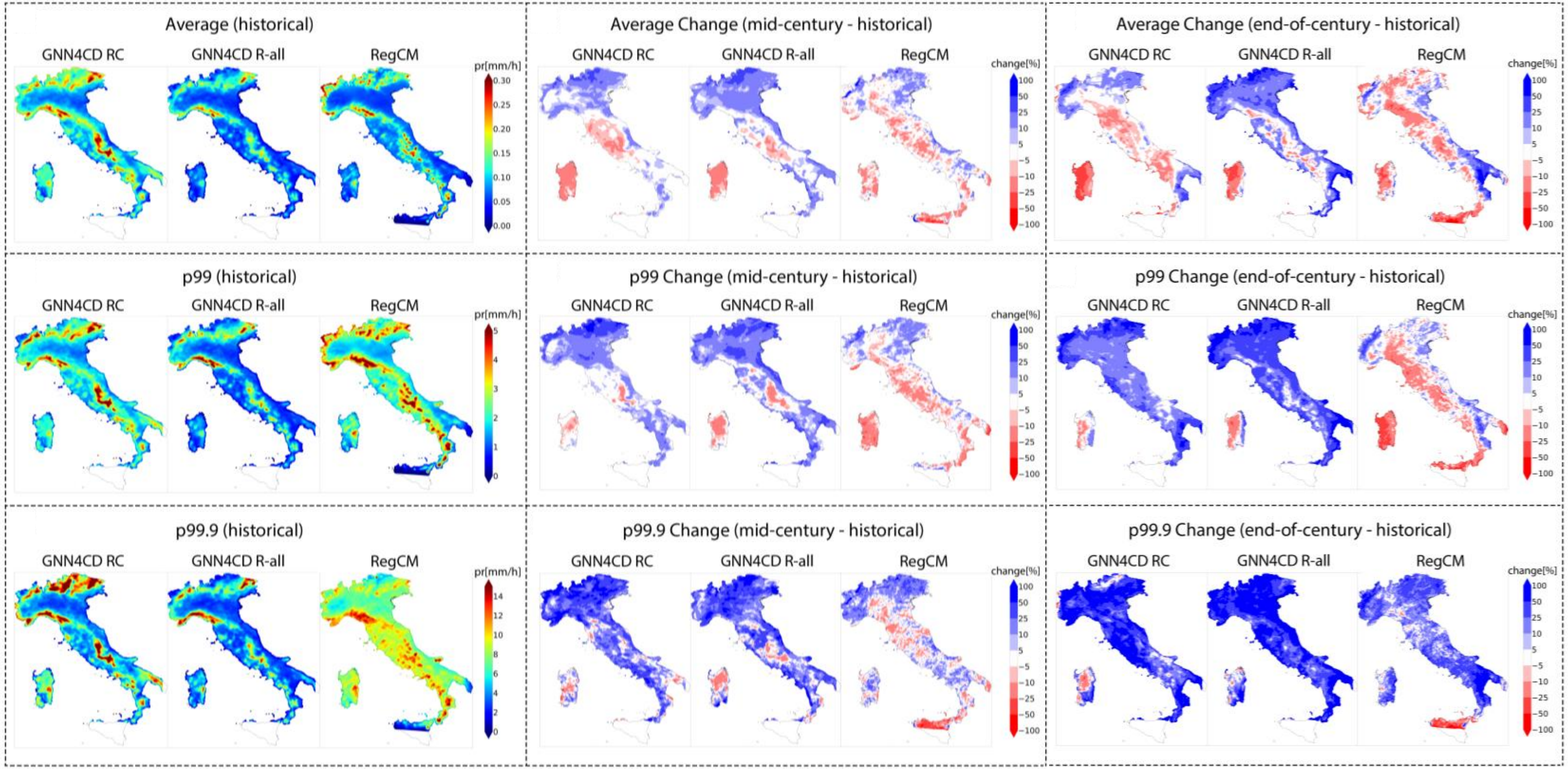
Emulation: R-all

Precipitation distribution



Emulation: RC and R-all

Historical | **Mid-century change** | **End-of-century change**



Conclusions

Next steps – GNN4CD model

- Improve the **R-all** model – one model is better than two!
- Improve the **spatial downscaling** (architecture) and the **distribution** estimation (loss/data)
- Estimate **uncertainty**
- Further investigate the **transferability** potential
- Use **additional variables**
- Explicitly address **climate change** impact on projections

Next steps – climate applications

- Retrain the emulator with **3h/6h** time resolution and **apply it to a broader collection of climate models simulations** → compare the **spread** of the emulator ensemble set of predictions with the spread intrinsic in the output of the climate numerical models

- Use the emulator in **other geographical areas** within the FPS-CORDEX domain (France Germany and Switzerland) without/with retraining

- Retrain the emulator to downscale **GCMs** simulations directly from ~**100 km to** ~**12 km**

Thanks!