Climate Informatics: Machine Learning for the study of Climate Change

Claire Monteleoni

University of Colorado
Boulder

October 2012: Hurricane Sandy – Reuters

October 2019: Kincade Fire, California – AFP

January 2014: Drought, Folsom Lake – California Department of Water Resources

November 2019: Flooding, Venice, Italy: Reuters

Machine learning can shed light on climate change.

Climate Informatics

2011 First International Workshop on Climate Informatics

2013 “Climate Informatics” book chapter [Monteleoni et al. 2013]

→ In the first 5 years: participants from over 19 countries and 30 U.S. states

2020 10th International Conference on Climate Informatics &
6th Climate Informatics Hackathon, September 23–26th *Oxford/Virtual*

→ Abstract and Paper submission deadline is TODAY!

Climate Change: Challenges for ML

[Banerjee & Monteleoni, Invited Tutorial, NeurIPS, 2014]

1. Past: Paleo-climate reconstruction

   What was the climate before we had thermometers?

2. Local: Climate downscaling

   What climate can I expect in my own backyard?

3. Future: Climate model ensembles

   How to reduce uncertainty on future predictions?

4. Spatiotemporal: Space and time

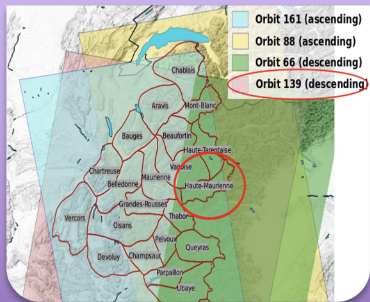
   How to capture dependencies over space and time?

5. Tails/impacts: Extreme events

   What are extreme events and how will climate change affect them?

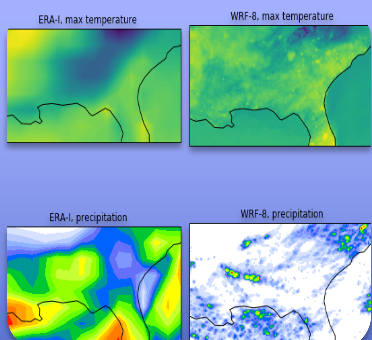
6. Other problems

   Data-rich playground with many opportunities for ML to have an impact!



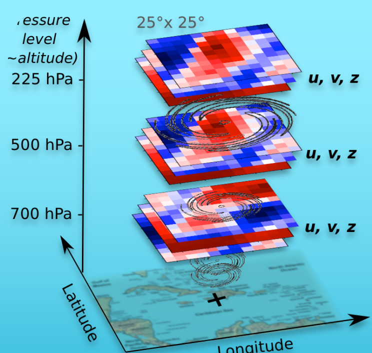
Semi-supervised DL

- Avalanche detection



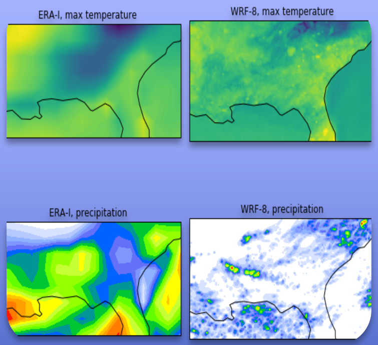
Unsupervised DL

- Temp. and precip. downscaling



Fused DL

- Hurricane track forecasting



{Un, Self}-supervised DL

- Temp. and precip. downscaling

Unsupervised Deep Learning

- Supervised DL. Prediction loss is a function of the label, y, and the network's output on input x.

Network output

$$f_W(x) = \hat{y}$$

Loss function

$$\mathcal{L}(\hat{y}, y)$$

- Unsupervised DL. Prediction loss is only a function of x, and the network's output on input x. There is no label, y.

Network output

$$f_W(x) = \hat{x}$$

Loss function

$$\mathcal{L}(\hat{x}, x)$$

Unsupervised DL for Downscaling

[Brian Groenke, Masters Thesis, CU Boulder, May 2020]
with help from Luke Madaus, Jupiter Intelligence

- Downscaling: Classic problem in climate & meteorology
  - Goal: use coarse-scale spatiotemporal data to infer values at finer scales
- Field of statistical downscaling, existing work:
  - Supervised learning methods
  - Provide point predictions
- Generative downscaling is largely open

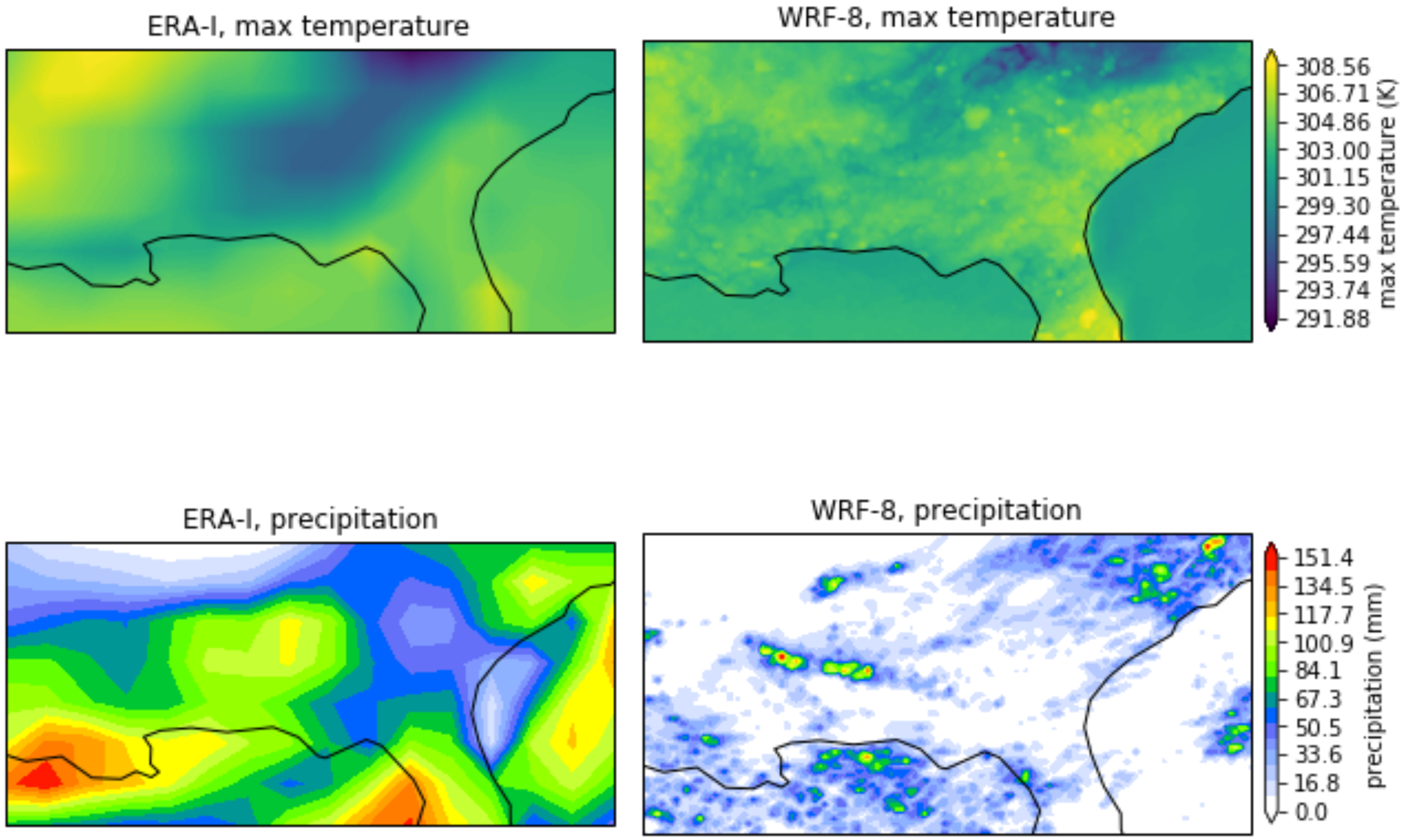
Unsupervised DL for Downscaling

[Brian Groenke, Masters Thesis, CU Boulder, May 2020]
with help from Luke Madaus, Jupiter Intelligence

- Cast downscaling as the ML task of domain alignment
- Extend deep unsupervised domain alignment
  - AlignFlow [Grover et al., AAAI 2020]
  - Glow normalizing flow [Kingma & Dhariwal, NeurIPS 2018]
  - Self-supervision via geographic alignment of both domains
- Obtain generative model for downscaling

Downscaling: training data

ERA: reanalysis data, 1° resolution;   WRF: numerical weather model prediction, $\frac{1}{8}$° resolution



Downscaling as domain alignment

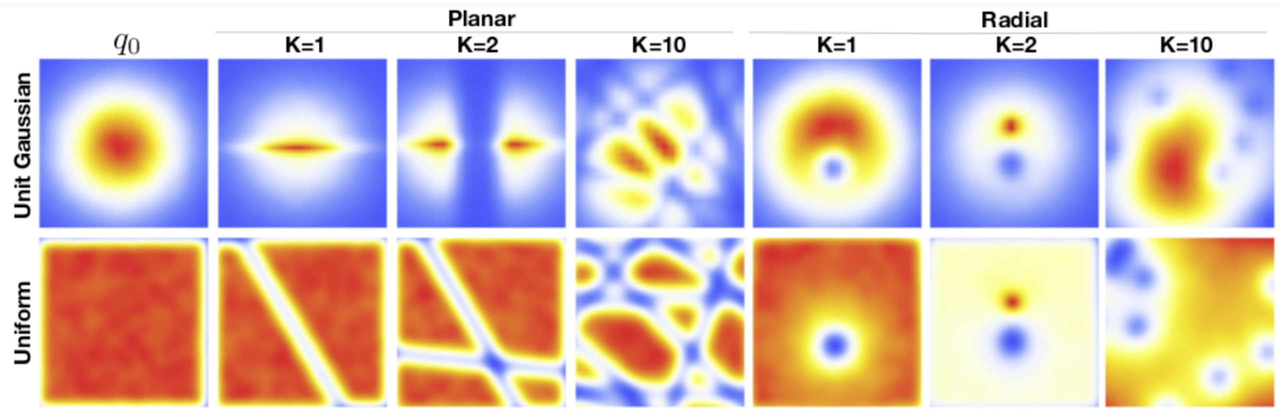
- Domain alignment task: given random variables X, Y, learn a mapping f: X → Y such that, for any x_i ∈ X and y_i ∈ Y,
  $f(x_i) \sim P_Y$ and $f^{-1}(y_i) \sim P_X$

- Downscaling as domain alignment
  - Learn the joint PDF over X and Y, by assuming conditional independence over a shared latent space Z

$$P_{XY}(x,y) = \int_{z \in Z} P_{XYZ}(x,y,z)dz \ = \int_{z \in Z} P(x|z)P(y|z)P_Z(z)dz$$

  - Model $P(x|z), P(y|z)$ using AlignFlow [Grover et al. 2020]
  - Starting with a simple prior on P_Z, learn a normalizing flow
  - No pairing between x and y examples needed!

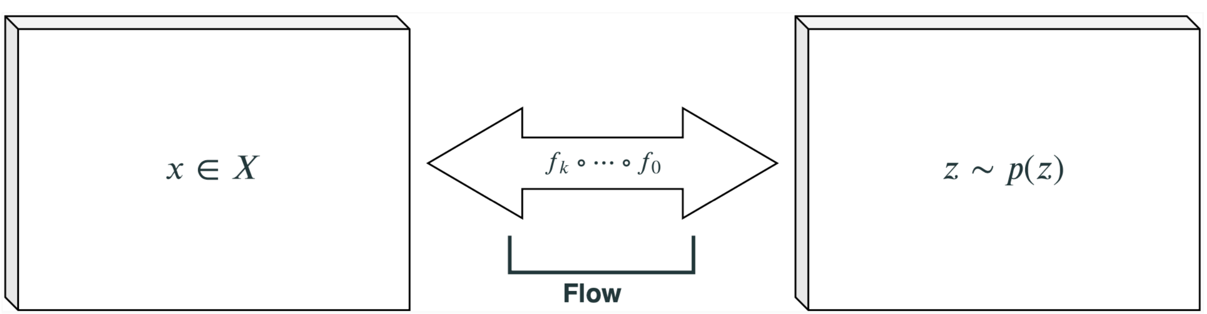
Normalizing Flows



[Rezende & Mohamed, 2015]

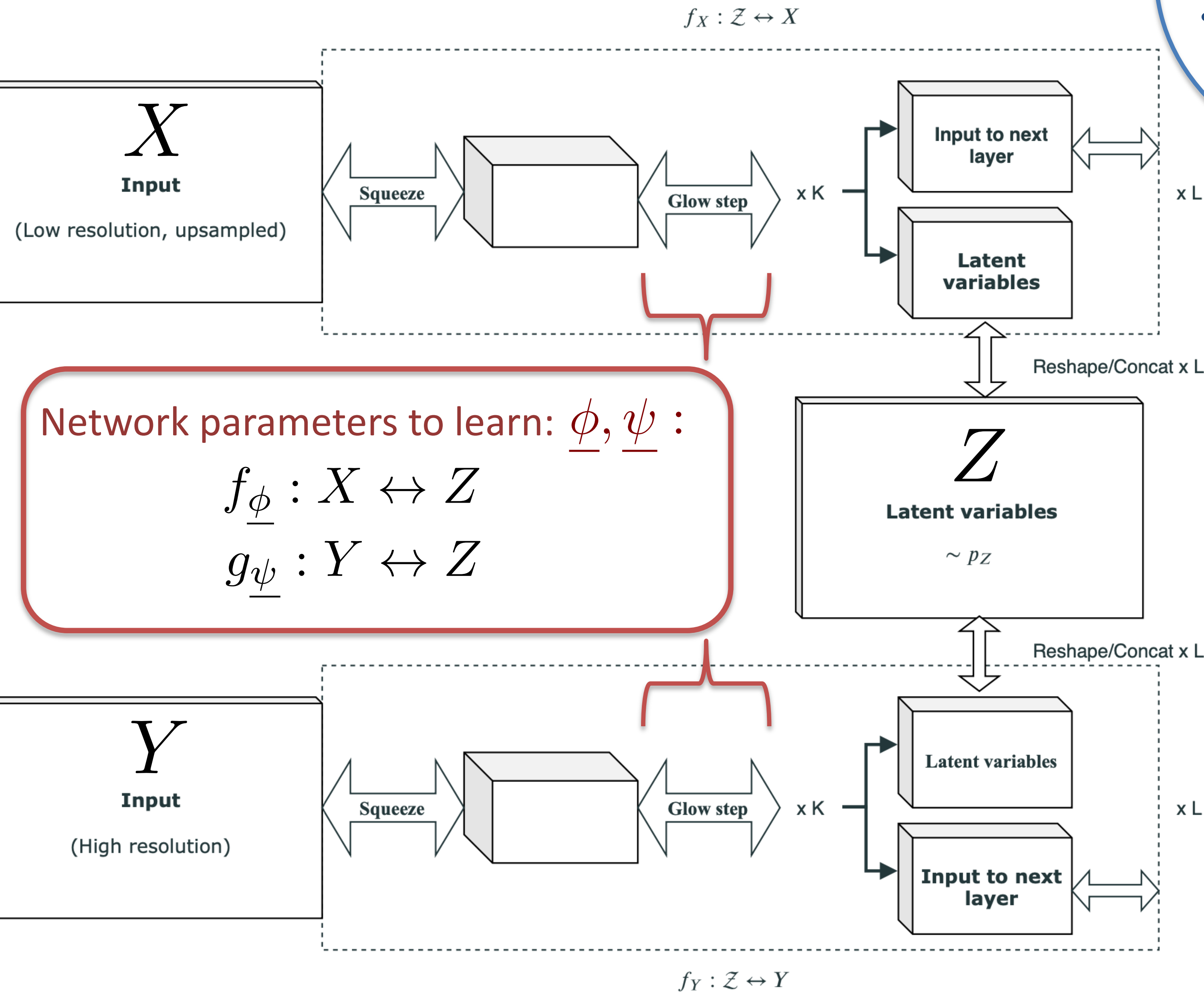
Learn a series of invertible transformations, $\{f_i\}$, from a simple prior on Z, to allow for more informative distributions on the latent space:

$$z_k = f_k \circ f_{k-1} \circ \cdots \circ f_1(z_0)$$



ClimAlign architecture

- Architecture follows AlignFlow [Grover et al., 2020]
- Normalizing flow: Glow [Kingma & Dhariwal, 2018]



Network parameters to learn: $\underline{\phi}, \underline{\psi}$:

$$f_{\underline{\phi}} : X \leftrightarrow Z$$

$$g_{\underline{\psi}} : Y \leftrightarrow Z$$

Comparison with supervised benchmarks

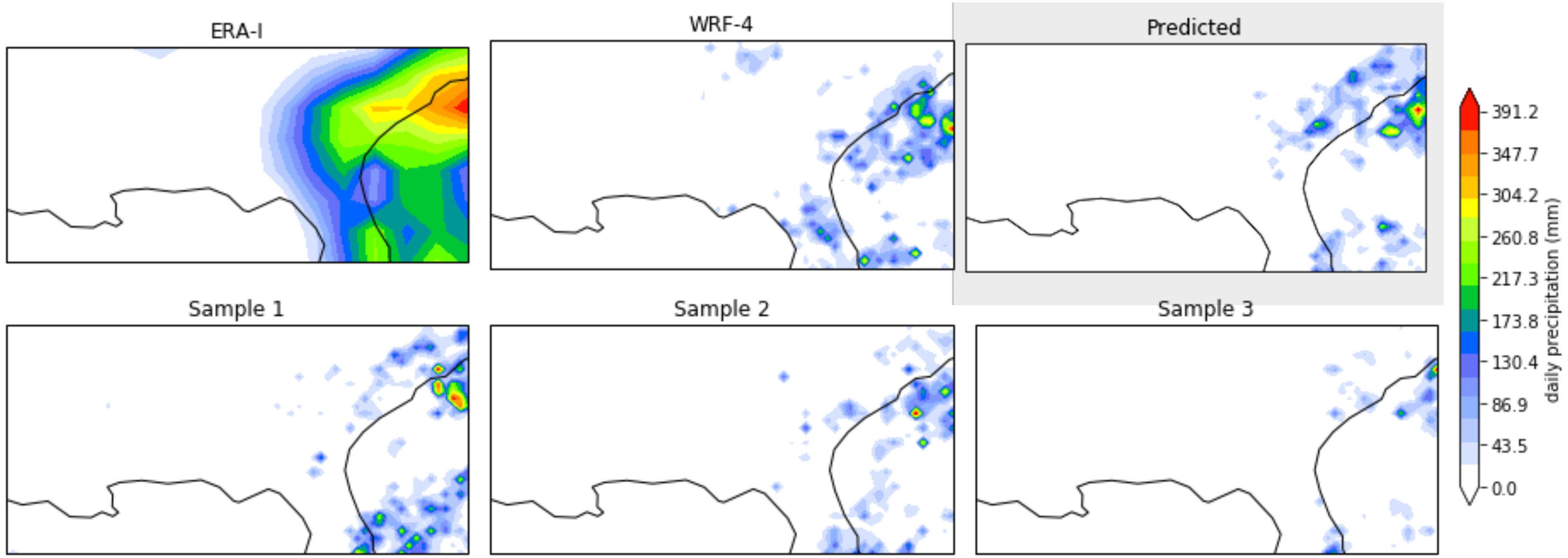
Temperature

Region	Method	RMSE	Bias	Corr
SE-US	BCSD	1.51 ± 0.15	-0.02 ± 0.21	$\mathbf{0.93} \pm 0.05$
	BMD-CNN	$\mathbf{1.30} \pm 0.12$	0.03 ± 0.13	0.90 ± 0.05
	ClimAlign (ours)	1.56 ± 0.13	$\mathbf{-0.005} \pm 0.22$	0.87 ± 0.06
P-NW	BCSD	1.54 ± 0.23	$\mathbf{0.01} \pm 0.10$	$\mathbf{0.95} \pm 0.03$
	BMD-CNN	$\mathbf{1.25} \pm 0.14$	-0.06 ± 0.05	0.93 ± 0.02
	ClimAlign (ours)	1.58 ± 0.18	0.03 ± 0.15	0.89 ± 0.04

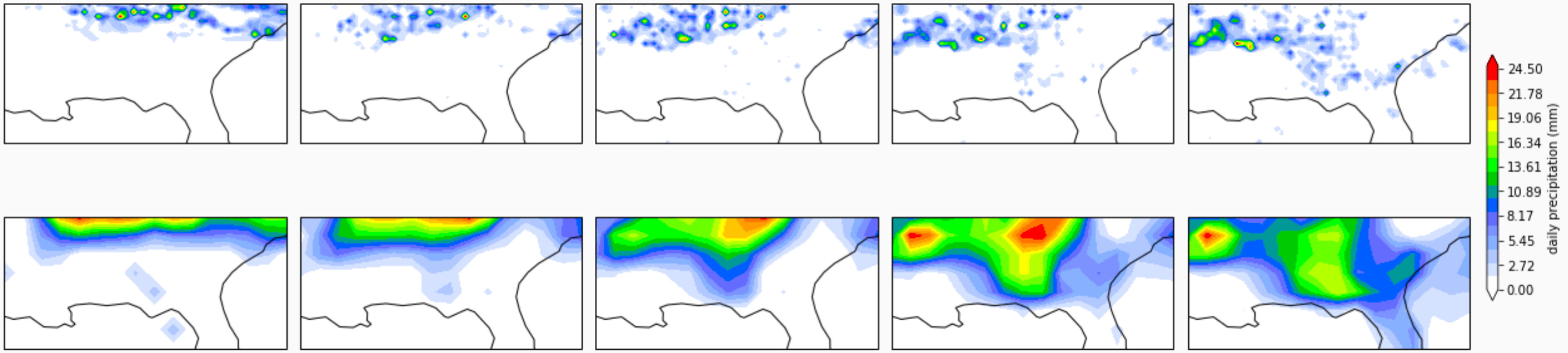
Precipitation

Region	Method	RMSE	Bias	Corr
SE-US	BCSD	27.32 ± 5.0	0.95 ± 1.4	0.39 ± 0.07
	BMD-CNN	$\mathbf{14.11} \pm 2.18$	-0.23 ± 0.47	$\mathbf{0.50} \pm 0.10$
	ClimAlign (ours)	18.40 ± 2.64	$\mathbf{0.08} \pm 0.86$	0.42 ± 0.07
P-NW	BCSD	8.90 ± 2.30	0.41 ± 0.26	0.61 ± 0.06
	BMD-CNN	$\mathbf{5.77} \pm 0.72$	$\mathbf{-0.18} \pm 0.61$	$\mathbf{0.70} \pm 0.03$
	ClimAlign (ours)	7.33 ± 0.69	0.54 ± 0.54	0.67 ± 0.03

Point prediction example



Interpolation example




Thank you!
Many thanks to:
Arindam Banerjee, University of Minnesota
Cécile Coléou, Météo-France & CNRS
Michael Dechartre, Irstea, Université Grenoble Alpes
Nicolas Eckert, Irstea, Université Grenoble Alpes
Sophie Giffard-Roisin, IRD Grenoble
Brian Groenke, University of Colorado Boulder
Anna Karas, Météo-France & CNRS
Fatima Karbou, Météo-France & CNRS
Luke Madaus, Jupiter Intelligence
Saumya Sinha, University of Colorado Boulder
cnrs dépasser les frontières
METEO FRANCE
Google Cloud
NSF
Boulder

Resources

- Climate Informatics: www.climateinformatics.org
  - Community network, data, resources, events

- 10th International **Conference** on Climate Informatics, September 2020, *Oxford/Virtual* ci2020.web.ox.ac.uk

- 9th International Workshop on Climate Informatics, 2019, Paris sites.google.com/view/climateinformatics2019

- Climate Informatics Hackathon: storm intensity forecasting
  github.com/ramp-kits/storm_forecast