Machine Learning meets Data Assimilation

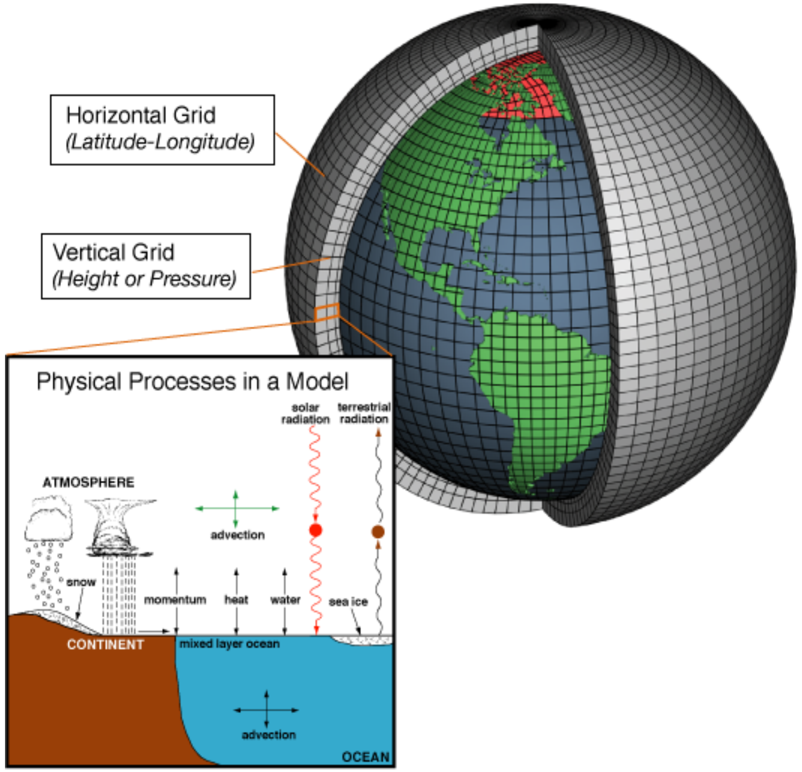
Peter Jan van Leeuwen

Peter.vanleeuwen@colostate.edu

Data Assimilation for NWP

Problem is how to provide best estimate and uncertainty estimates of physically related high-dimensional fields, e.g. global NWP, or high-resolution regional NWP.

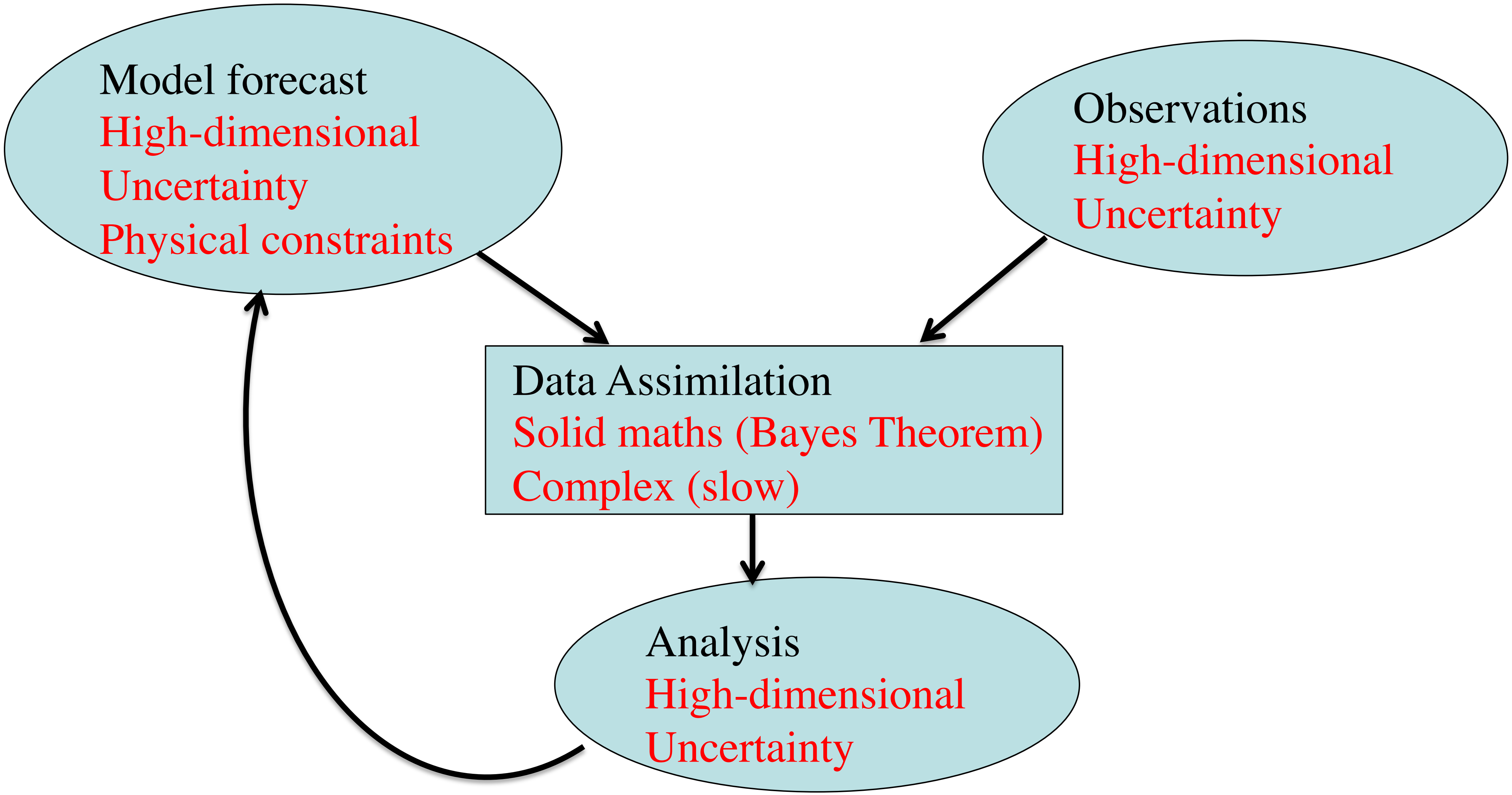
This is different from 'down-scaling' in which only one or a few variables are to be estimated, typically from model forecast and observations.



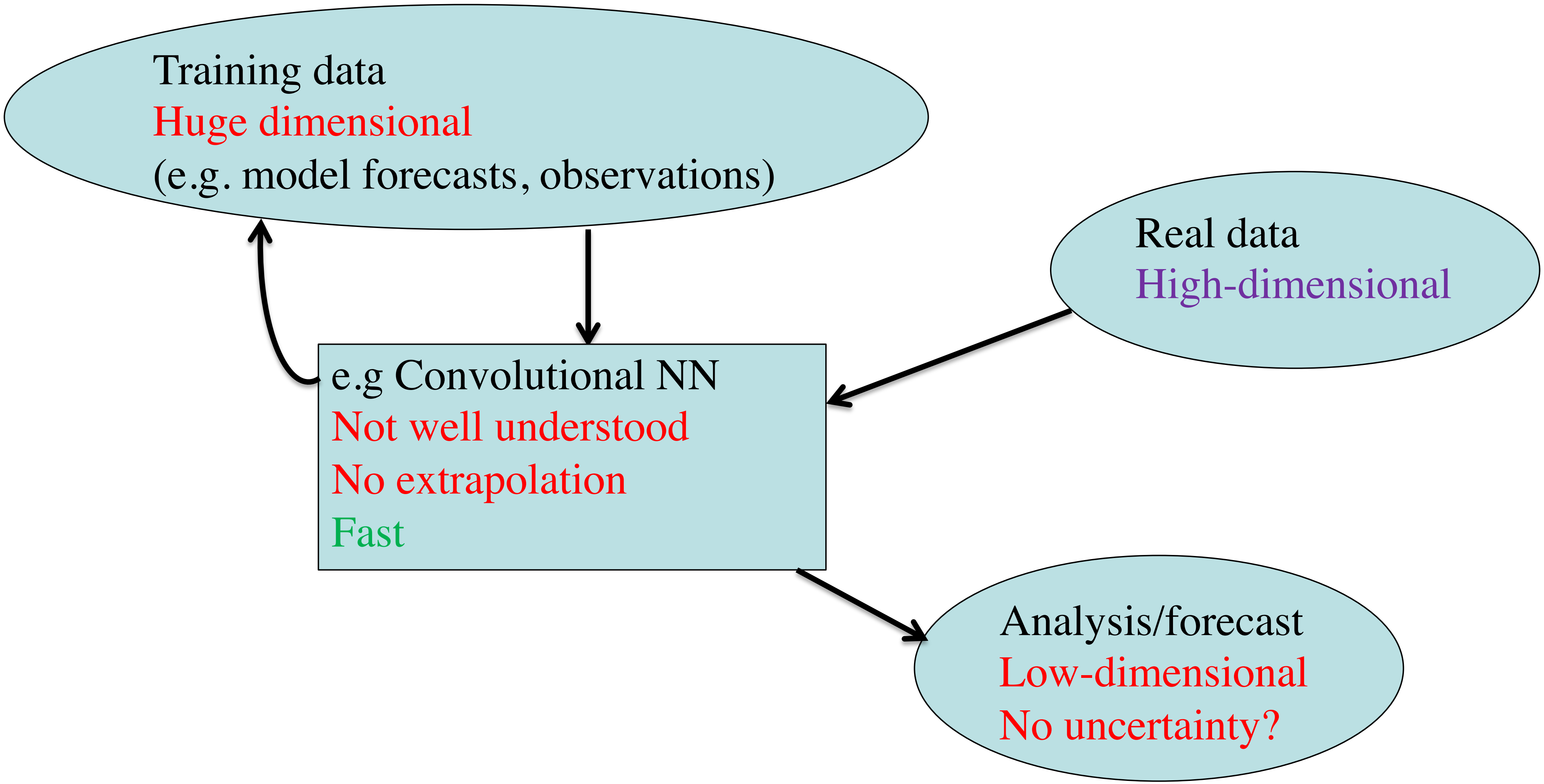
.

Photo by Brian Cook on Unsplash

Data Assimilation for NWP



Deep Learning for NWP?



Deep learning: Uncertainty quantification

Deep learning provides a nonlinear map from input to output :

$$y = f(x) + \eta$$

DA is not a discrete problem, so uncertainty quantification is relatively easy.

The noise term η can be estimated from the test data (DL).

UQ for the output ***y*** can be determined from known uncertainty in the input ***x*** via sampling and propagation through the network:

$$y + \delta y = f(x + \delta x) + \eta$$

We know how to do this… !

Deep learning: Extrapolation

Deep learning minimizes a costfunction

$$J(\mathbf{w}) = \sum_i \frac{1}{r_i}(y_i^{NN} - y_i^{obs})^2 + \lambda \mathbf{w}^T \mathbf{w}$$

Good for input within space spanned by training data, bad for extrapolation.

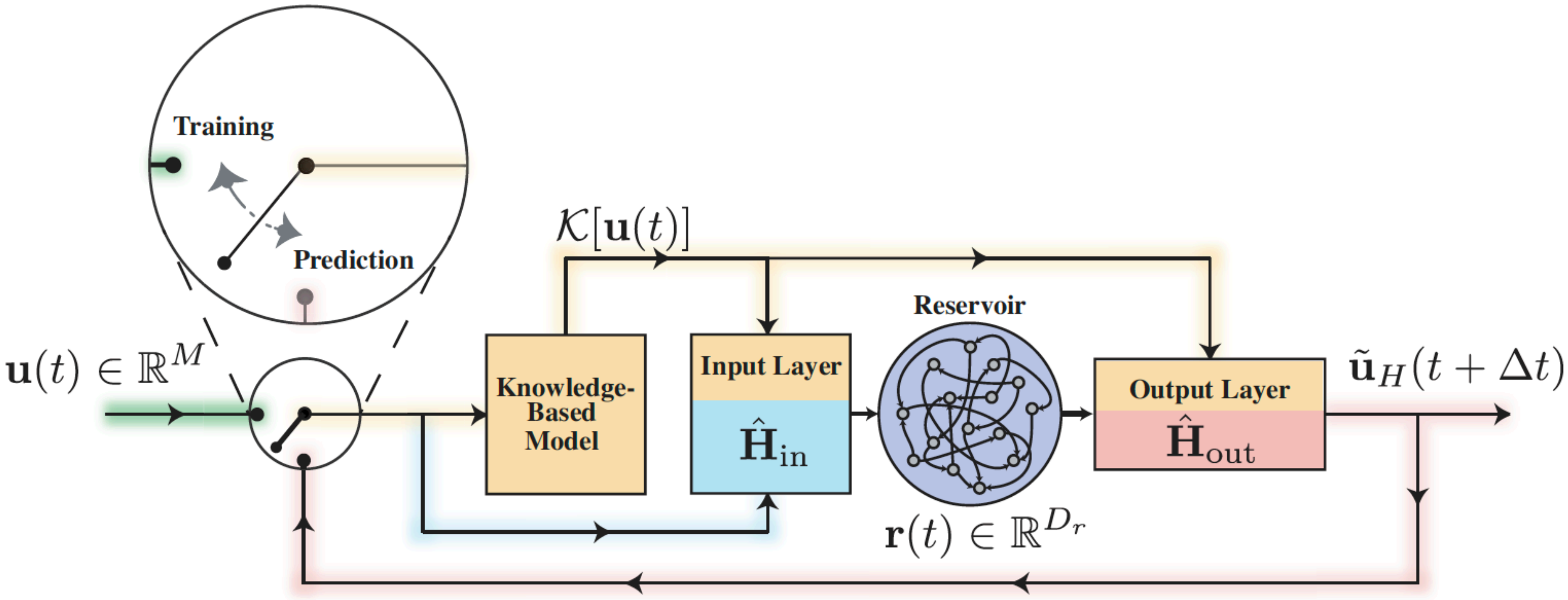
Bring in physical constraints between output variables:

$$J(\mathbf{w}) = \sum_i \frac{1}{r_i}(y_i^{NN} - y_i^{obs})^2 + \lambda \mathbf{w}^T \mathbf{w} + \mu g(\mathbf{y}^{NN})$$

Starts to look like Data Assimilation, e.g. 3DVar !

Deep learning: Extrapolation

Build physics-based forecasts into costfunction, e.g. Reservoir computing (ML) and physics-based model:



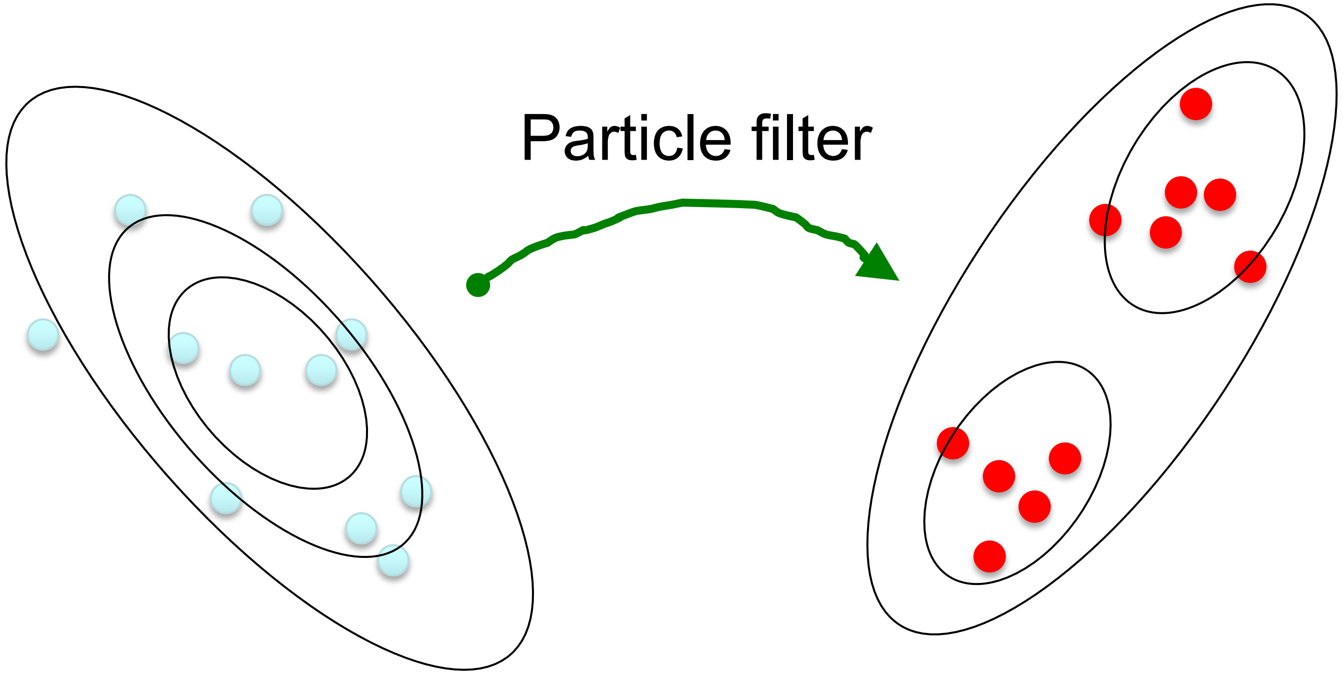
(Pathak et al., 2018)

Explore ML techniques within DA

Powerful ML techniques:

- Kernel embeddings
- Stochastic gradient descent
- Deep learning
- Etc ...!

Nonlinear Data Assimilation (Retrievals)



Ensemble of model states
before observations

Ensemble of model states
after observations

Particle flows

The particles, so full atmospheric states, are propagated in artificial time *s* via

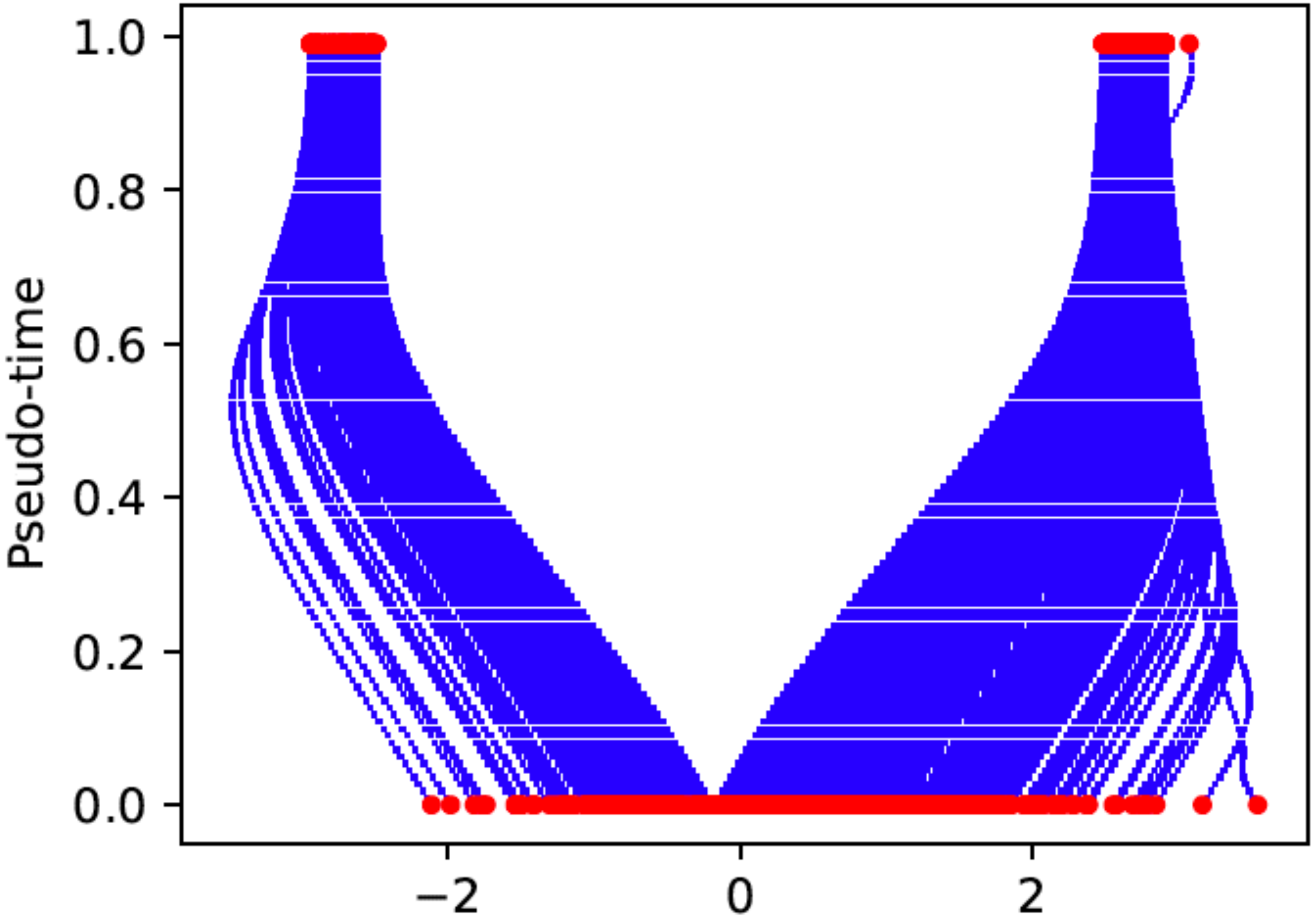
$$\frac{dx}{ds} = f_s(x)$$

The question is: How to choose the vector flow field $f_s(x)$?
Need an efficient algorithms that iteratively decrease the distance between the pdf of the particles and the posterior pdf (solution to Bayes Theorem).

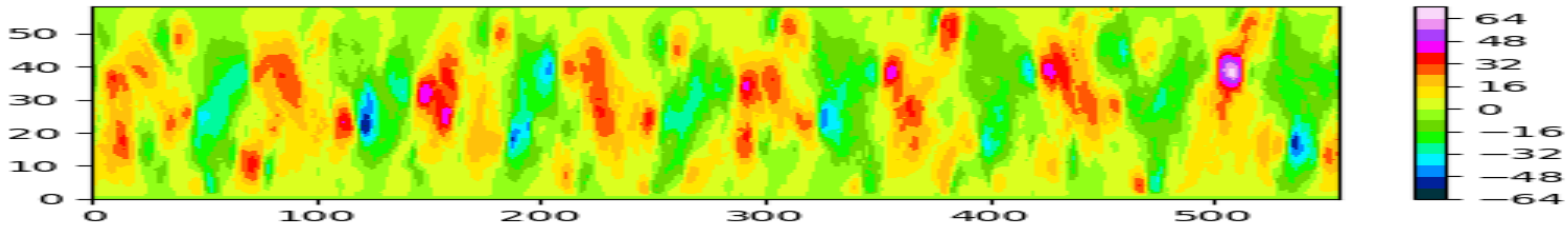
Use kernel embedding (ML) of the flow field:

$$f_s(x) = \langle K(x,\cdot), f_s(\cdot)\rangle_{\mathcal{F}}$$

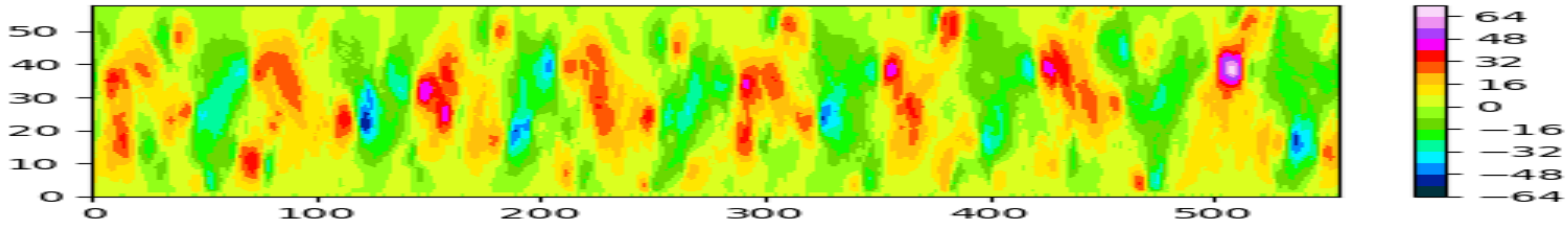
Particle flow in action



Results exploring ML in DA



True meridional velocity



Particle mean meridional velocity

Stochastic gradient descent (ML) in DA

We tried stochastic gradient descent methods like ADAM, but found difficulties with physical balances.
Instead we used an approximate Newton method:

$$A^{-1}\Delta x = -\nabla dist$$

Convergence accelerated when we add momentum memory as in RMSProp.
First calculate momentum update:

$$\mathbf{v}^{n+1} = \beta \mathbf{v}^{n} - (1-\beta) A \nabla dist$$

Then update move in state space as:

$$\Delta x = \eta \mathbf{v}^{n+1}$$

Conclusions

- Data assimilation has firm basis in Bayes Theorem.
- Deep Learning can benefit strongly from DA techniques
- DA and ML should merge, within the Bayesian framework.
- ML techniques are extremely useful for nonlinear DA, e.g. kernel embedding, (elements of) stochastic gradient descent.
- More work needed to merge more completely.
- High-resolution ocean and atmospheric DA exploring ML techniques is underway.

8th International Symposium on Data Assimilation

Canvas Stadium, Colorado State University

Fort Collins, Colorado, USA

8th – 12th June, 2020