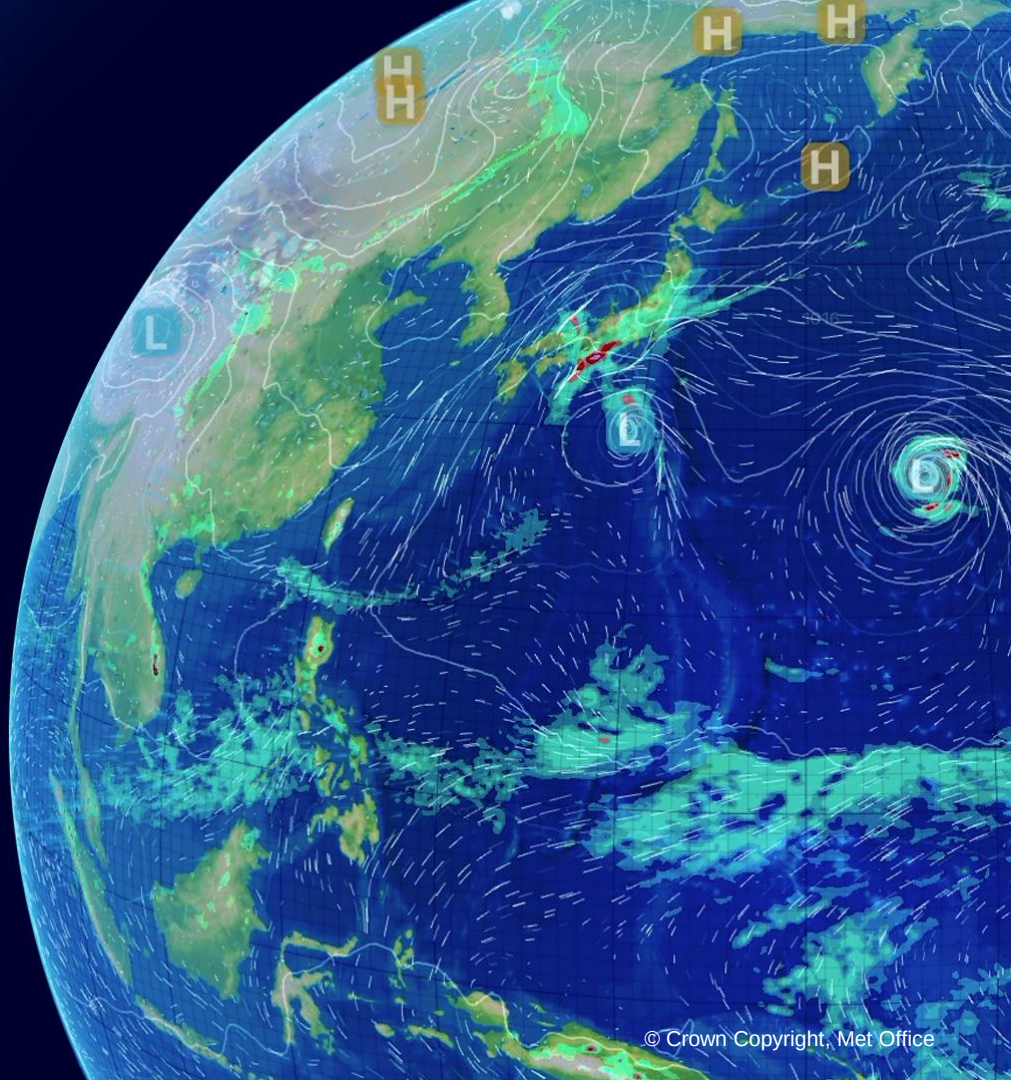
# Review of Machine Learning activities for model development

Keith Williams



# Types of ML for model development

- Emulation of model functions
- Parametrization emulation
- Parametrization replacement
- Model replacement

Increasing risk & potential benefit

- Elements of data assimilation

- [Post processing – see hidden slides]

# Emulation of model functions

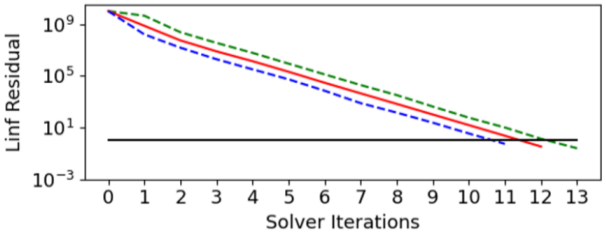
# Numerical weather forecasts: To precondition the linear solver

- Linear solvers are important to build efficient semi-implicit time-stepping schemes for atmosphere and ocean models.
- However, the solvers are expensive.
- The solver efficiency depends critically on the preconditioner that is approximating the inverse of a large matrix.

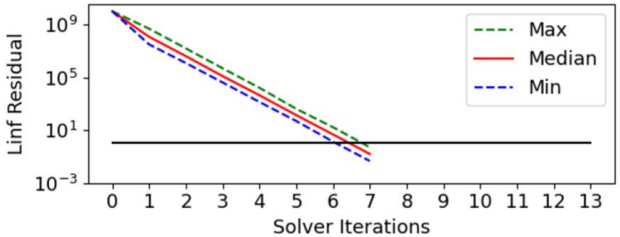
**Can we use machine learning for preconditioning, predict the inverse of the matrix and reduce the number of iterations that are required for the solver?**

**Testbed:** A global shallow water model at 5 degree resolution but with real-world topography.
**Method:** Neural networks that are trained from the model state and the tendencies of full timesteps.

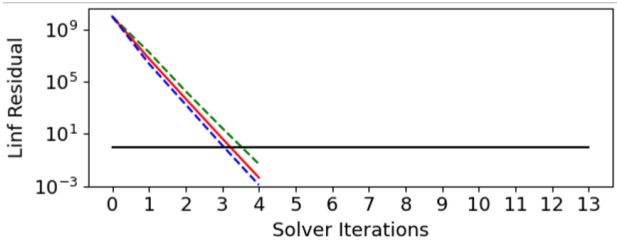
**No preconditioner:**




**Machine learning preconditioner:**




**Implicit Richardson preconditioner:**




**It turns out that the approach (1) is working and cheap, (2) interpretable and (3) easy to implement even if no preconditioner is present.**

Ackmann, Dueben, Smolarkieicz and Palmer https://arxiv.org/abs/2010.02866

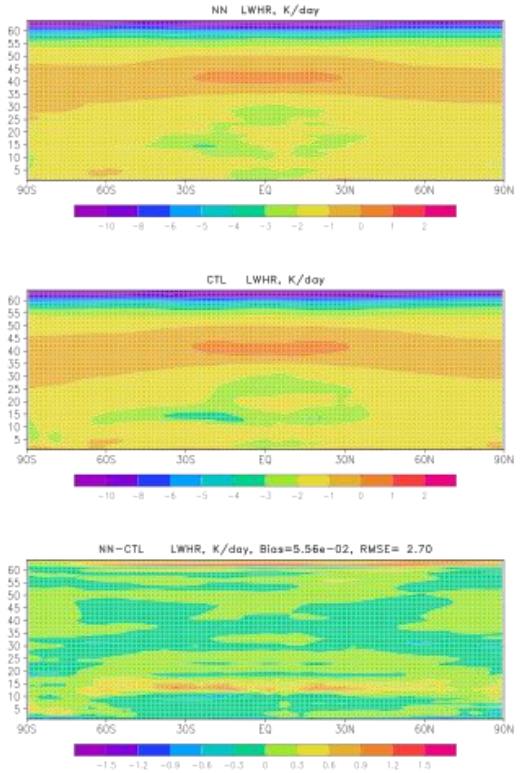
# Parametrization emulation & optimal parameter selection

# Machine Learning at CMC

- Ongoing investigation of a genetic algorithm for optimal parameter selection in the global ensemble data assimilation system (manuscript submitted to Monthly Weather Review)
- Initiation of formal collaboration between CMC and Mila – Quebec AI Institute to investigate parameterization emulators:
  - Significant resources invested under 3-year special project funding (two CMC researchers and two Mila researcher)
  - Initial target is optimization of the Li and Barker radiation scheme using either full emulation or a hybrid ML-Gauss-Legendre Quadrature (QLG) approach

# Robust NN Emulators of Radiation Schemes

- Accurate, fast, and stable NN-based emulators of radiative transfer parameterizations were built using 17 years worth of output from 2011 version of NCEP CFS, a fully coupled atmosphere-ocean-land-ice state-of-the-art climate model with time-varying radiative forcings, capturing diurnal, annual, and decadal variability, both internal and forced.
- In the intervening decade, the atmospheric model, GFS, underwent a complete overhaul: dynamical core, many physics parameterizations, and software infrastructure were replaced, all other components were updated.
- Nevertheless, radiative transfer emulators developed almost a decade ago are surprisingly robust with respect to substantial structural and parametric change in the host model: when used in the AMIP-like experiment with the new GFS, they not only remain stable, but generate realistic output.

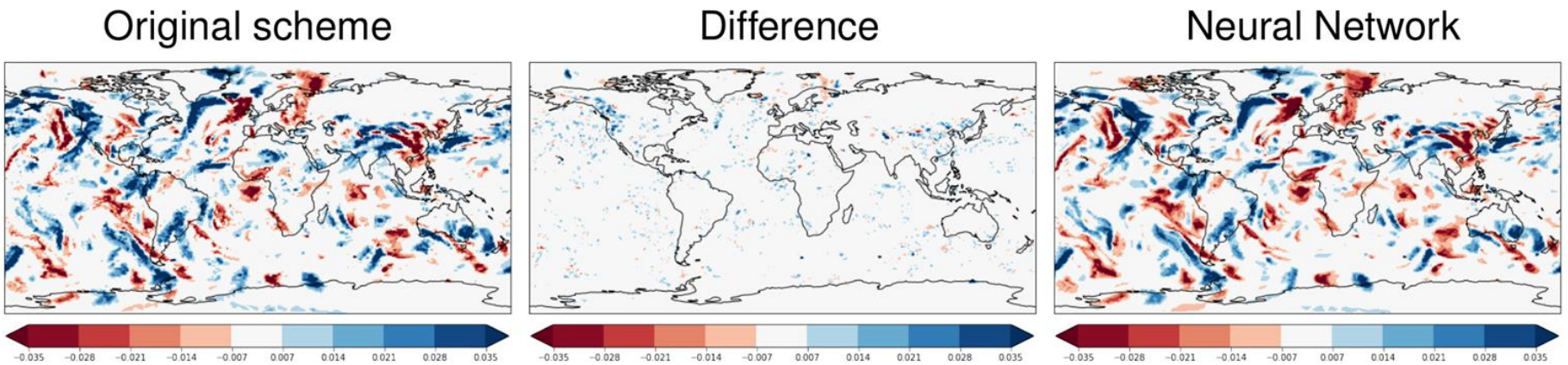



LW and SW Heating Rates, K/day, averaged over 12 months AMIP-like run

# Numerical weather forecasts: To emulate gravity wave drag

- Repeat the same approach for the gravity wave drag scheme of IFS
- Start with non-orographic and continue with orographic wave drag

**Results for the non-orographic gravity wave drag are promising.**




**There is also a nice relation between network size and accuracy.**

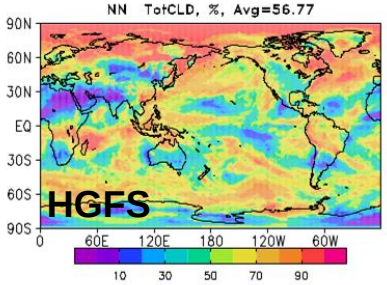
**However, it is still questionable whether computational performance of the Neural Nets is better when compared to the conventional scheme.**

**Results are not as good for the orographic gravity wave drag scheme.**

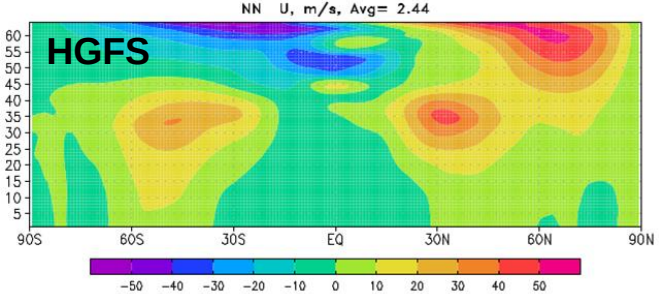
Chantry, Dueben, Palmer

**The emulator was used successfully to generate tangent linear and adjoint code within data assimilation.**

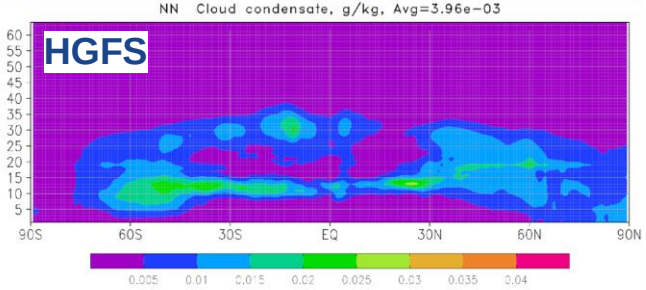
# 2020 Pilot Project: Hybrid GFS (Physical Dycore + ML physics)



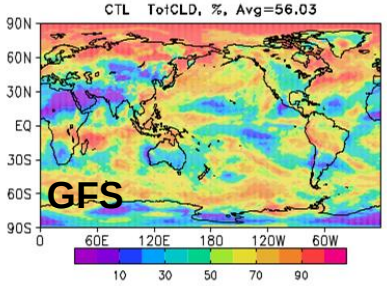

Bias = 0.74%



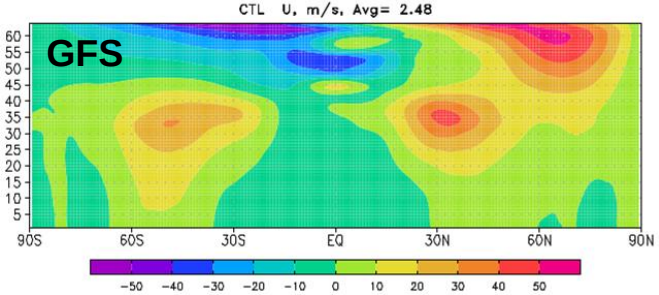

Bias = 0.04 m/s



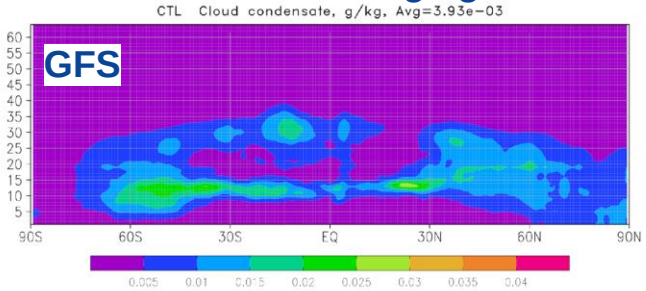

Bias = 3.e-5 g/kg




Total Cloudiness (vertical mean)




U-component of wind (zonal mean)




Cloud condensate (zonal mean)

**Validation:** 24 parallel runs (10-day GFS v.16 forecasts each), uniformly covering entire 2018
*In all 24 runs no signs of instability were observed!*

# Parametrization replacement

# Predicting Marine Fog: Machine Learning of Atmospheric Boundary Layer Turbulence

U.S. NAVAL RESEARCH LABORATORY

David Flagg, Jeff Byers, Katarina Doctor, Tommy Jensen, Saša Gaberšek, Jim Doyle

**Q:** ***Can we use large-eddy simulation (LES) to improve our ability to parameterize eddy diffusivity ($K_H$) and eddy viscosity ($K_M$) in numerical weather prediction (NWP)?***

Typical NWP definitions: $K_H, K_M =$
$f$( mixing length scale, turbulence kinetic energy (TKE), . . . )

## Concept

Use supervised machine learning to build a formula for **K** as a function of quantities relevant to turbulent eddies, trained on LES data to approximate the real atmosphere

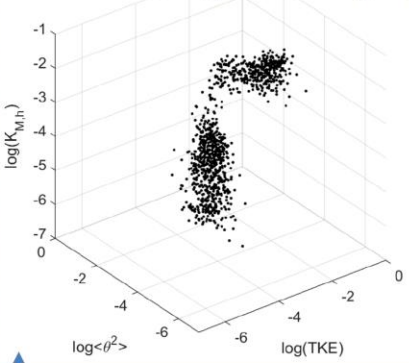
Local K−theory closure: $\overline{u_j'\phi'} = -K \frac{\partial \bar{\phi}}{\partial x_j}$

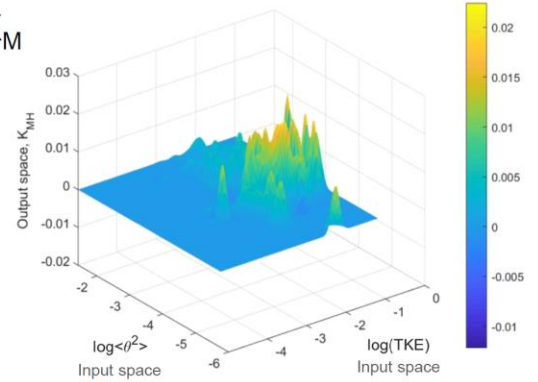
From **LES:** $K = \frac{-\overline{u_j'\phi'}}{\frac{\partial \bar{\phi}}{\partial x_j}}$ and $K = \boldsymbol{f}(\boldsymbol{x})$

$\boldsymbol{x}$ = quantities relevant to turbulent diffusion

Gaussian Process Regression (GPR): $\boldsymbol{y} = \mathbf{K}(\boldsymbol{x})$

## Initial Testing: Deriving $K_H$, $K_M$

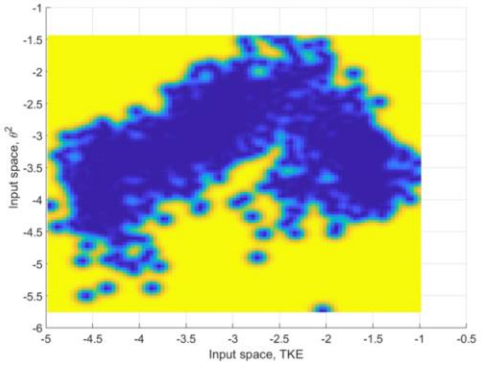



**Random sample of LES space with $K_H$, $K_M$ and two predictors: TKE and $\overline{\theta'^2}$**

**Predicting $K_H$, $K_M$ from trained GPR model ($10^4$ points)**

**GPR model variance (statistical coverage); Yellow = high variance, insufficient input data**



# Machine Learning the Warm Rain Process

Gettelman et al 2020, in revision for JAMES
https://www.essoar.org/doi/10.1002/essoar.10503868.1

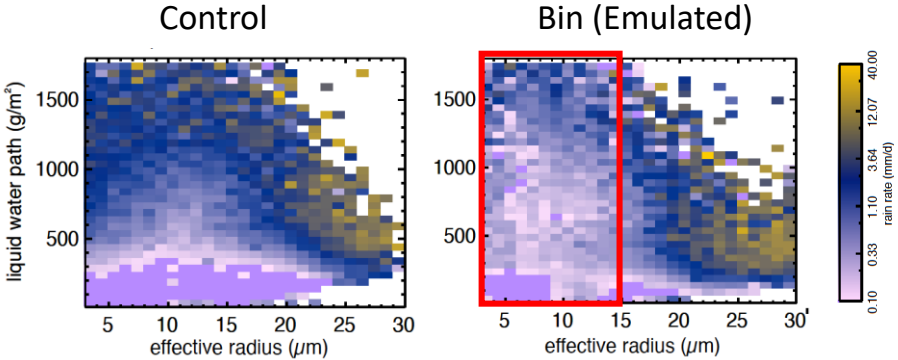
Can we do the warm rain process better?
Replace autoconversion, accretion and self collection
Use a stochastic collection kernel from a bin code

1. Break distributions of cloud and rain into bins
2. Run stochastic collection kernel from a bin microphysics code
3. Use altered distributions to estimate AUTO+ACCRE tendency

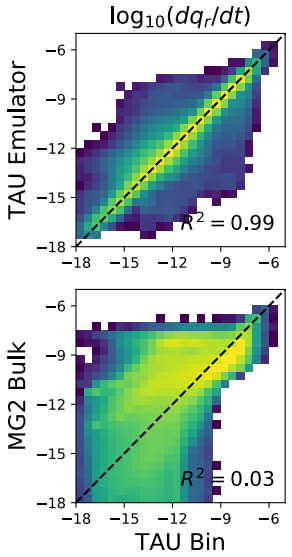
Results:

We can change the answer in the model with the bin code. Very slow. Interesting results.
Recover speed and recover results with a neural network emulator
Embedded NN in the microphysics: maintains conservation with series of checks

Control =significant rain for all $R_e$
Bin = Little rain for $R_e$<15um
Bin looks like an LES model




Emulator Performance




Different than original model

# ML/Fortran Integration Pilot Project

Integrating trained ML components back into Fortran-based Weather and Climate codes is not trivial.

Review of current methods:

- *Re-coding specific model architectures into Fortran*.
- Pros: fast, good compatibility with existing Fortran codes.
- Cons: time-consuming and inflexible as changing the ML model architecture means recoding in Fortran.
- *Calling Python from within Fortran (e.g. using a wrapper library such as CFFI)*
- Pros: No re-coding as the ML model remains in Python.
- Cons: This approach was tried in the Met Office and we found various compatibility issues on our HPC. It is also potentially slow since Fortran needs to call out to Python.
- *Use a Pure Fortran NN library, or bridging library (there are several existing solutions e.g. FANN, neural-fortran, FKB, frugally-deep)*
- Pros: fast (probably), good compatibility with existing Fortran codes.
- Cons: The existing solutions only have limited architectures and algorithms available since they are either Fortran re-implementations of methods or APIs that mirror common ML frameworks, Translations from native ML model format are often required.

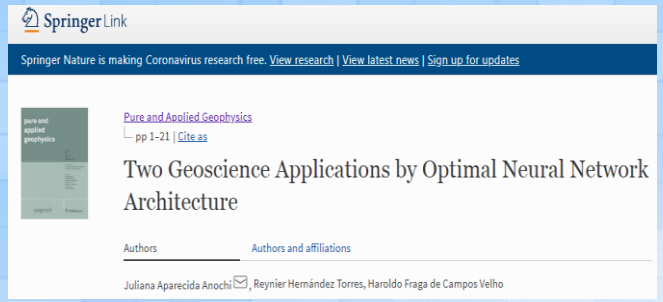
New approach: to interface directly to the underlying C/C++ libraries of the ML frameworks using Tensorflow Lite C/C++ API

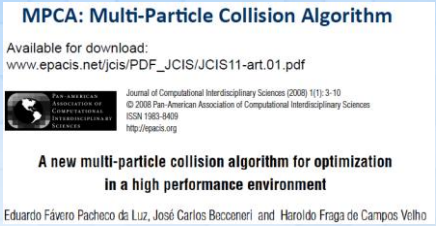
# Model replacement

# AI applied for Seasonal climate forecast

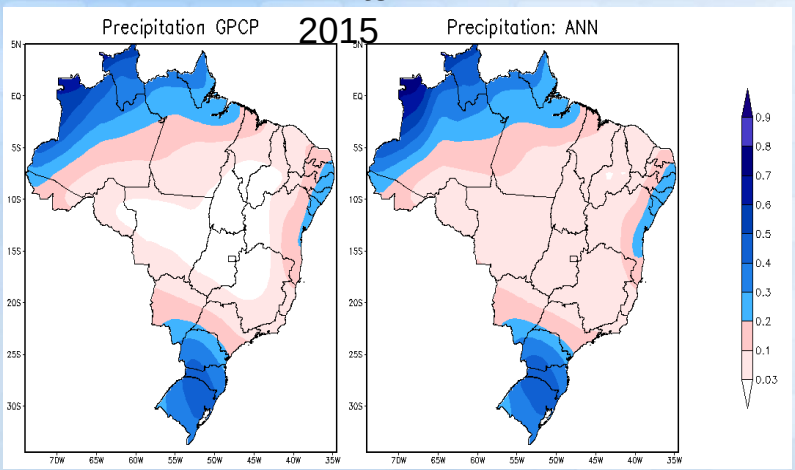
Research developed at INPE/Brazil
Juliana Anochi, researcher at INPE
juliana.anochi@inpe.br

Autoconfigured ANN using Multi-Particle Collision Algorithm

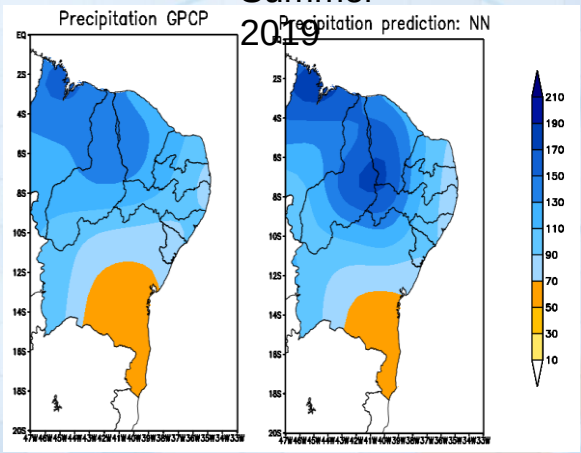



Winter 2015



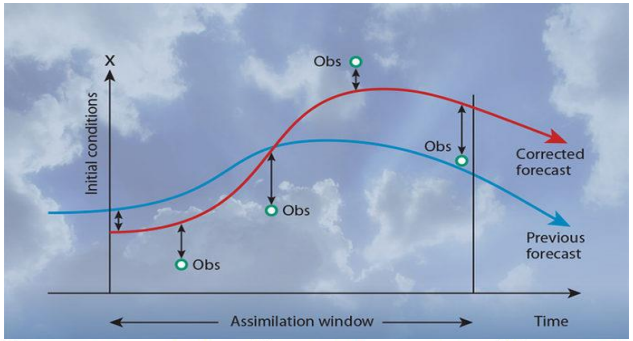

Summer 2019



# Data assimilation

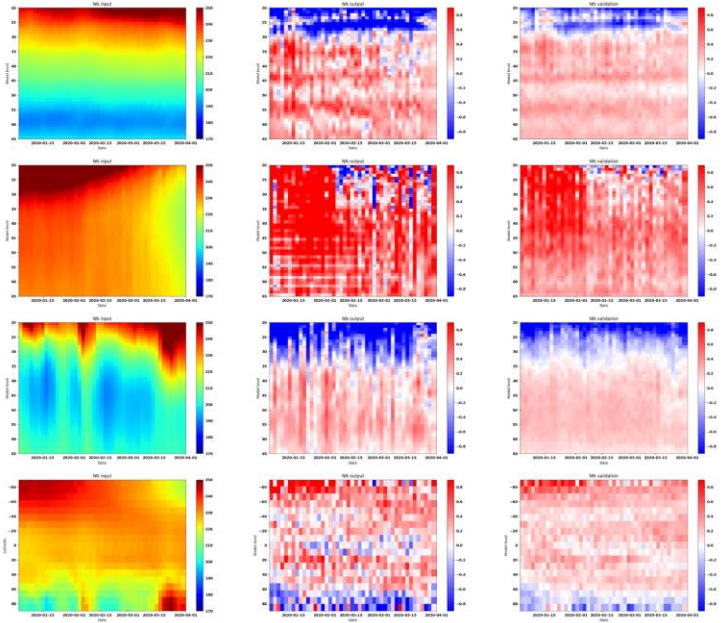
# Data assimilation: Bias-correct the forecast model in 4DVar data assimilation

- Data-assimilation blends observations and the forecast model to generate initial conditions for weather predictions
- During data-assimilation the model trajectory is “synchronised” with observations for the same weather regimes
- It is possible to learn model error when comparing the model with (trustworthy) observations




**Two approaches:**

- Learn model error within the 4DVar data-assimilation framework for so-called “weak-constraint 4D-Var”
- Learn model error from a direct comparison of the model trajectory to observations or analysis increments using deep learning (column-based or with three-dimensional)



**Benefit:**

When the bias is learned, it can be used to:

- Correct for the bias during data-assimilation to improve initial conditions
- Correct for the bias in forecast simulations to improve predictions (discussed controversially)
- Understand model deficiencies

**Patrick Laloyaux, Massimo Bonavita** and Peter Dueben @ ECMWF + Thorsten Kurth and David Matthew Hall @ NVIDIA

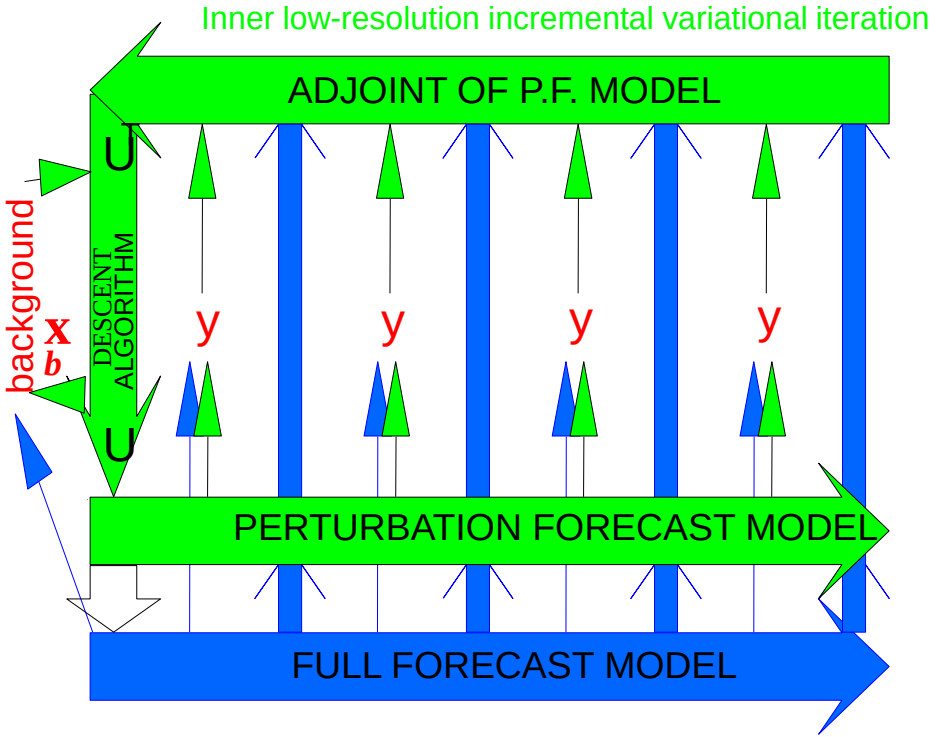
# Deep learning the tangent linear model in 4D-Var

The tangent linear / perturbation forecast (PF) model has two functions in 4D-Var data assimilation:

- To evolve a perturbation of the model state through the assimilation window.
- To form an adjoint for the gradient descent algorithm.

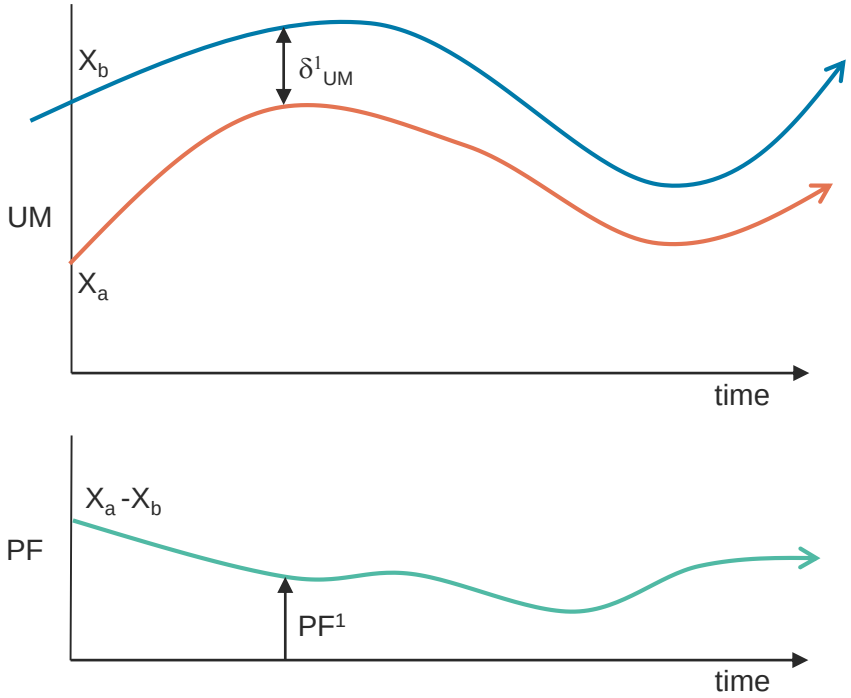
Creating the PF model physics components analytically is time consuming and difficult.

**Can deep learning be used to create data-driven PF model components?**



# Deep learning the tangent linear model in 4D-Var

- Training data generated by running full UM in pairs (1 perturbed, 1 control) for 1 PF model time step.
- Difference between UM runs, $\delta^1_{UM}$, is used as ‘truth’.
- Learning target can be $\delta^1_{UM}$ or…
- PF model error, $PF^1 - \delta^1_{UM}$
- PF model can also be run in dynamics only mode



# Conclusions

- Types of ML for model development:
  - Emulation of model functions
  - Parametrization emulation
  - Parametrization replacement
  - Model replacement

Increasing risk/benefit
(Probably easier wins near the top of the list)

- Elements within DA also promising.
- Radiation still popular to emulate. Is it actually the best option?
- Promising work on calling ML from Fortran using C/C++ interface.

# Questions

# Discussion

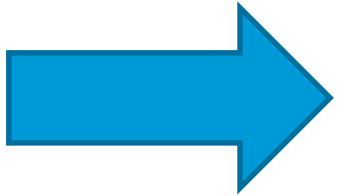
Comprehensiveness and realism of the training data set used during development of AI/ML model components along with a synergistic collaboration between both ML and modeling experts are important factors contributing to:

- Generalization capability of the ML component and stability of the model utilizing it.
- Ability of the ML component to remain functional without re-training despite changes in the host model.

# NN Full Suite of Atmospheric Physics

**Atmospheric Physics Suite**
**(GFS v16, C96L64):**

LW Radiation,
SW Radiation,
Planetary BL,
Orographic and convective gravity wave drag,
Deep convection,
Shallow convection,
Microphysics,
$CO_2$(t), trace gases,
Aerosols (tropo- and stratospheric),
$O_3$ and $H_2O$ photochemistry



**NN**
**522 Inputs**
**304 Outputs**
**250 Hidden Neurons in one hidden layer**

***3 times faster***

Belochitski A. and V. Krasnopolsky, 2020: Emulation of a Full Suite of Atmospheric Physics Parameterizations in NCEP GFS using a Neural Network, *ECMWF-ESA Workshop on Machine Learning for Earth System Observation and Prediction.*

**Training:** Data simulated by 24 10-day GFS v.16 forecasts, uniformly covering entire 2018 (*radiation was calculated at each physics time step!*).

# Discussion

- A proof-of-concept NN emulator of an entire suite of atmospheric physics parameterizations in a state-of-the-art CGM was built using a training data set capturing diurnal and seasonal cycles in a year's worth of model generated data. The emulator was validated in 24 ten day forecasts with initial conditions spanning a full year.
- Preliminary results show that application of methodology, successfully used previously in development of accurate, fast, and stable NNs for radiative transfer, results in a stable and fast emulator of the entire physics suite that provides satisfactory accuracy of model simulation.

# AI at Météo-France

- "Machine learning is being used to tune the post-processing ensemble probabilities in order to maximize their value for end users (see publication about thunderstorm prediction at doi:10.1080/16000870.2019.1696142). This approach has been applied for multi-ensemble probabilistic forecasts of thunderstorms and rain accumulations. The algorithm is an optimization of observation-based forecast scores such as ROC error statistics, it is a complement to conventional calibration techniques."

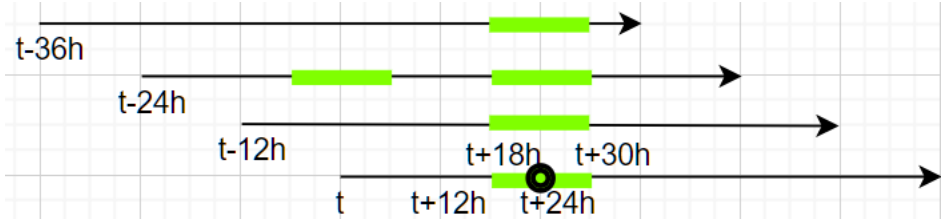
- "It is planned to apply a data mining technique (association mining) to long archives of past ensemble predictions in order to detect weather regime-dependent forecasting system biases, such as a lack of ensemble dispersion, in order to define priorities for research on improving NWP systems."

- "Use of U-Net Convolutional Neural Network (CNN) for detection of meteorological structures out of the NWP models : bow echoes, mid-latitude fronts, tropical cyclones.

# COSMO-Ru ML postprocessing at Hydrometcentre of Russia (RHM)

The COSMO-Ru model postprocessing is applied to t2m, td2m, PMSL, wind 10m, gusts.

1. Bias correction (at SYNOP stations points) – *in operational mode*
2. Additional artificial neural network (ANN) based correction using several last forecasts (see Fig.) at SYNOP stations points (including up to 3 lagged forecasts) – *in operational mode*




The ANN processed 24-h forecast takes into account the bias correction output for several lead and initial time intervals (green)

3. Interpolation of correction increments (**1**&**2**) to **a grid** – *under development*

The 2017-2020 training dataset for each atmospheric parameter contains **~2×10$^8$** forecast-SYNOP observation pairs. ANN is trained on the direct model output for **four COSMO-Ru** model **configurations** with **13.2, 6.6, 2.2, 1.0km grid spacings**.

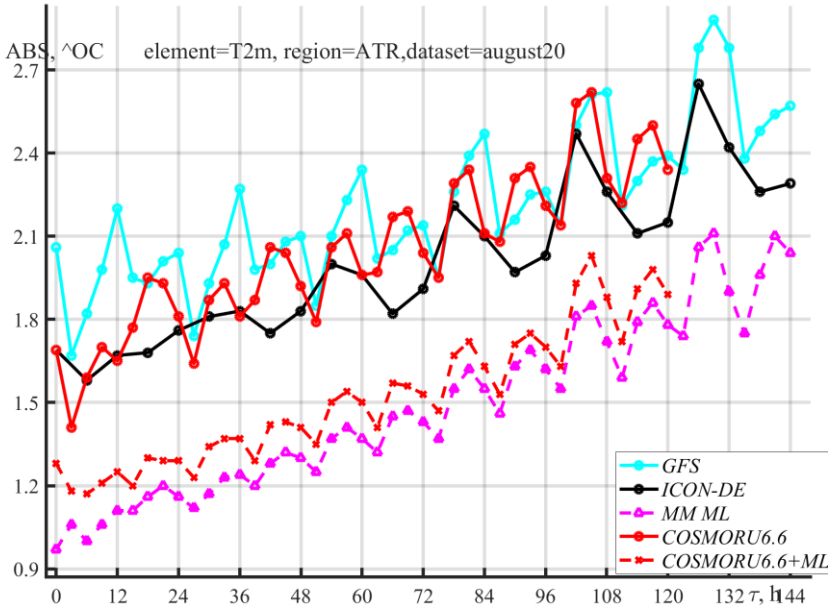
Bykov, F.L. Statistical Correction of the COSMO Model Weather Forecasts Based on Neural Networks. Russ. Meteorol. Hydrol. 45, 141–152 (2020). https://doi.org/10.3103/S1068373920030012

# *Testing ML postprocessing at RHM*

- The COSMO-Ru6.6-ENA ML processed forecasts have the accuracy close to the accuracy of raw forecasts with **2-4 days shorter lead time** depending on the geographical region

- The complex multimodel (MM) ML processed forecast takes into account COSMO-Ru6.6-ENA, GFS, ICON, SL-AV forecasts. The accuracy of MM ML forecasts is close to the accuracy of COSMO-Ru6.6-ENA ML processed forecast **with ~1 day shorter lead time**

Testing results for t2m in Asian Russia in august 2020



# *PP-MILEPOST: MachIne LEarning-based POST-processing*

Main contributors:

IMGW-PIB (A. Mazur, G. Duniec, J. Linkowska), RHM (F. Bykov, G. Rivin, A. Bundel), MCH (D. Cattani, D. Nerini)

Plans (2020-2022):

- Review and survey of the history of ML methods and state-of-the-art in the world
- Development and testing of new methodologies for selecting an appropriate subset of predictors
- Participants **collect and exchange the** historical **datasets** of the predictors and the target values
- Explore artificial neural networks (ANN) performance optimizations for CPU/GPU
- Tuning ANN's hyper-parameters and architectures
- Develop ML suggested package(s) to be **shared among participating members**