HELMHOLTZ AI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Helmholtz-Zentrum hereon

Fortran-Python Bridge for Machine Learning Applications in Earth System Models

Caroline Arnold

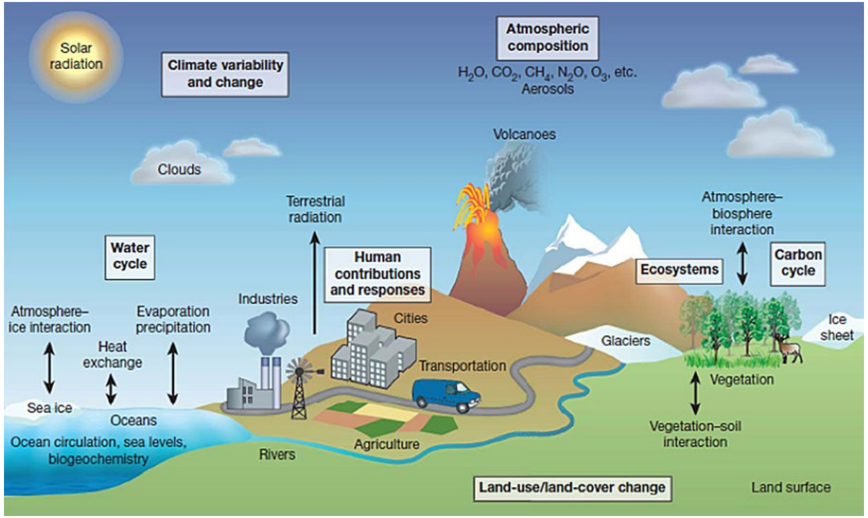
6th Workshop on Coupling Technologies
Toulouse, 20.01.2023

Machine Learning (ML) in Earth System Models

Typical scenarios

- Sub-grid processes, e.g., cloud microphysics, atmospheric chemistry, …
  - Described by parameterizations
  - Neglected due to computational effort

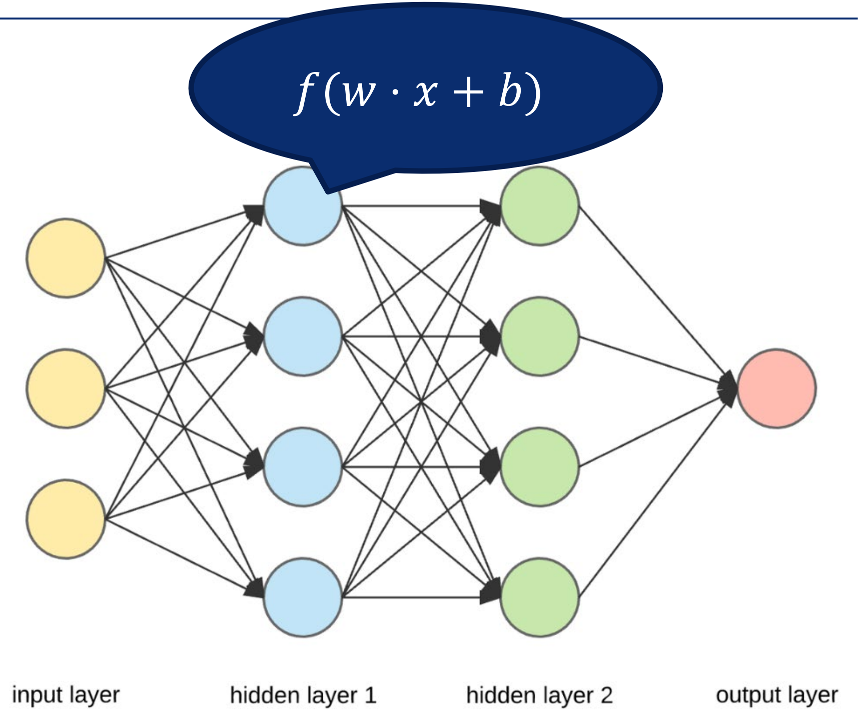
- Machine learning (ML) algorithm emulates sub-grid process
- → Can provide more accurate process description



Machine Learning

A very brief overview

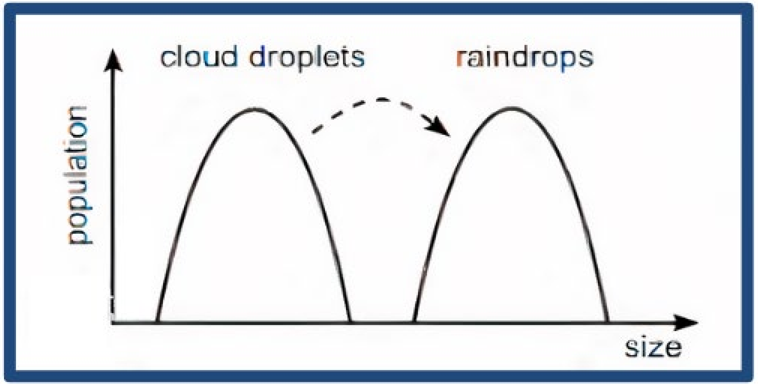
- Statistical algorithms that learn from data
- Most relevant: neural networks
- Training a ML model
  - Optimize for a given metric („loss function“) – accuracy, mean squared error, …
  - Adjust model parameters w, b to best fit the training dataset
  - Save model for later use



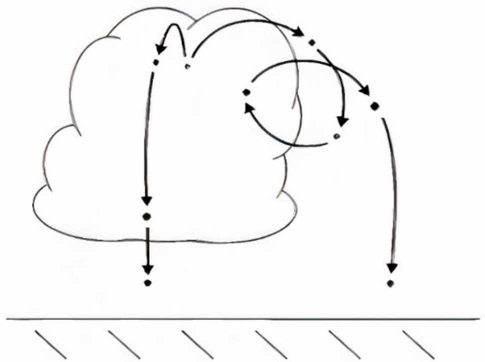
Machine Learning for cloud microphysics in ICON-NWP

Example for „learning“ a scheme

- Clouds and rain represented by water content and droplet concentration („2-moment-scheme“)
- Updates by bulk-moment scheme



- ML training data: super-droplet scheme
- → more accurate, less assumptions



Source: Ann-Kristin Naumann, MPI-M

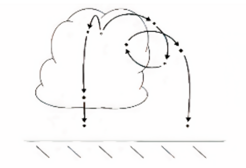
Machine Learning

„Offline“ and „online“ model evaluation

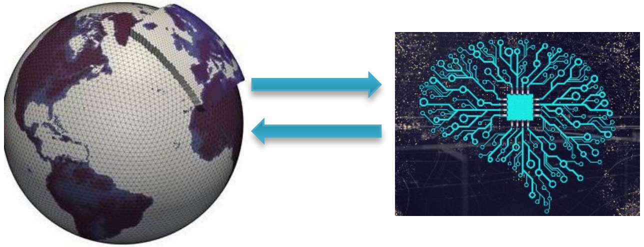
- How does the ML model generalize to unseen data?

→ Evaluation on test dataset

- Report metrics and publish



- But in reality …
  - ML model applied in ESM time loop
  - New conditions encountered
  - Interaction with the ESM

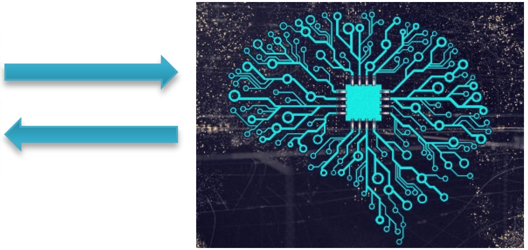


Good „offline“ performance does not necessarily imply good „online“ performance!

Bridge to integrate ML model in ICON

Quick and flexible „online“ tests

- **flexible**
  - Allow for iterative development of ML model
- **ML-developer-friendly**
  - Limit changes to ESM code
- **performant**
  - Runtime overhead should not limit development
- **accelerated**
  - Use ML specialised hardware (GPU) if possible

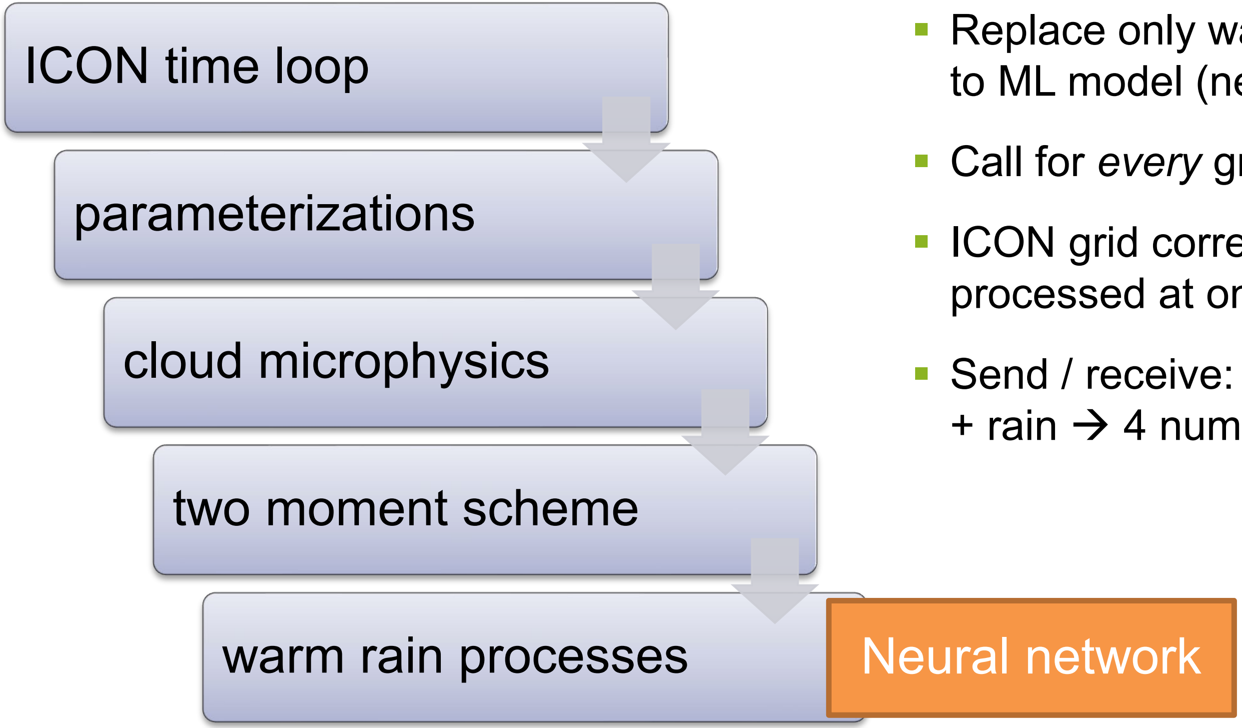


ICON (Fortran)

ML model (Python)

ICON program flow

ML model for warm rain microphysics



- Replace only warm rain processes by call to ML model (neural network)
- Call for *every* grid cell and vertical level
- ICON grid corresponds to ML batch – processed at once
- Send / receive: 2 moments each for cloud + rain → 4 numbers

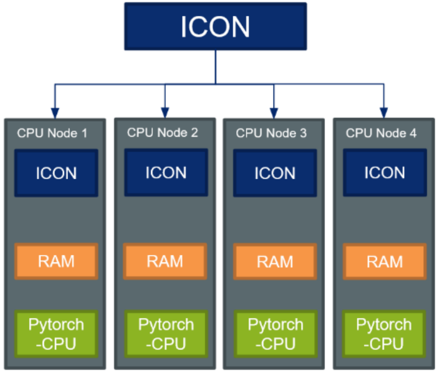
HELMHOLTZ AI

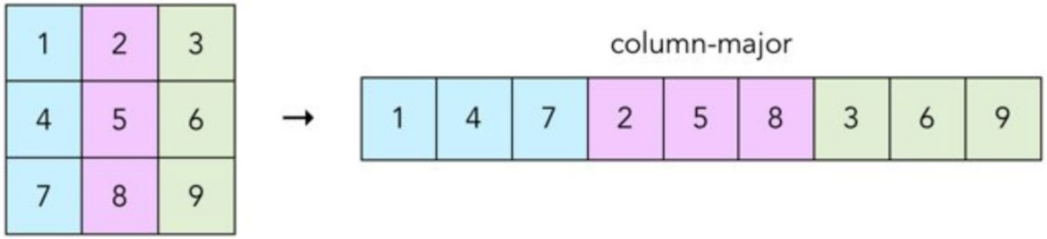
ICON – Python bridge 1

„Embedded“ Python using C Foreign Function Interface (CFFI)

- CFFI compiles Python code to dynamic library
- linked to ICON at compile time
- Py interpreter initialized at ICON runtime
- Executes frozen Py code locally

- Data transfer:
  - Memory address transmitted
  - Read from / write to buffer
- Beware of column-major (F) / row-major (C) order! → swap dimensions





HELMHOLTZ AI

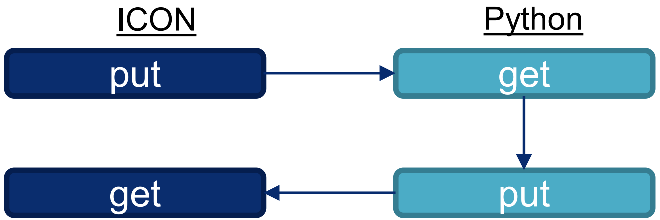
ICON – Python bridge 2

Using Yet Another Coupler (YAC)

- Can we use existing coupling tools to run the ML model in ICON?
- Use the Python bindings for YAC and the new coupling setup in ICON
- No interpolation
- We currently need one exchange field for each vertical level

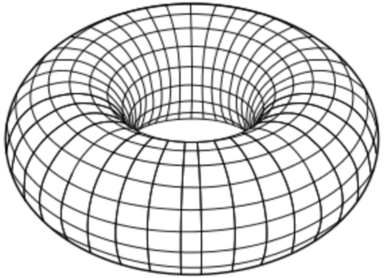
→ Demo case only for a simple scenario

- MPI Communicator splitting



Test scenario: warm bubble

Comparing different ICON – Python bridges



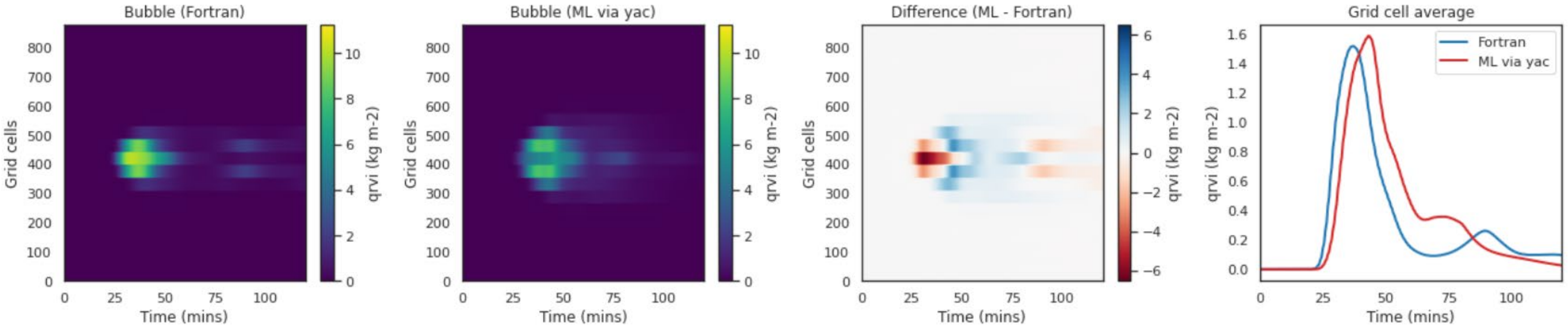
- Torus grid (20 x 44 cells with PBCs)
- 70 vertical levels (atmosphere)
- Focus on formation of one cloud
- High resolution
- High temperature to prohibit ice formation and focus on warm rain processes
- Suitable for testing the ML model
  - Time step: 20 seconds
  - Simulation time: 2 hours (360 steps)

Temperature (°C)

Applying the ML model

Warm bubble scenario

- Compare the vertically integrated rain rate
- ML model trained on super-droplet scheme to replace bulk-moment scheme
- Cooperation with climate scientists to verify sanity
- We can exchange the ML model easily

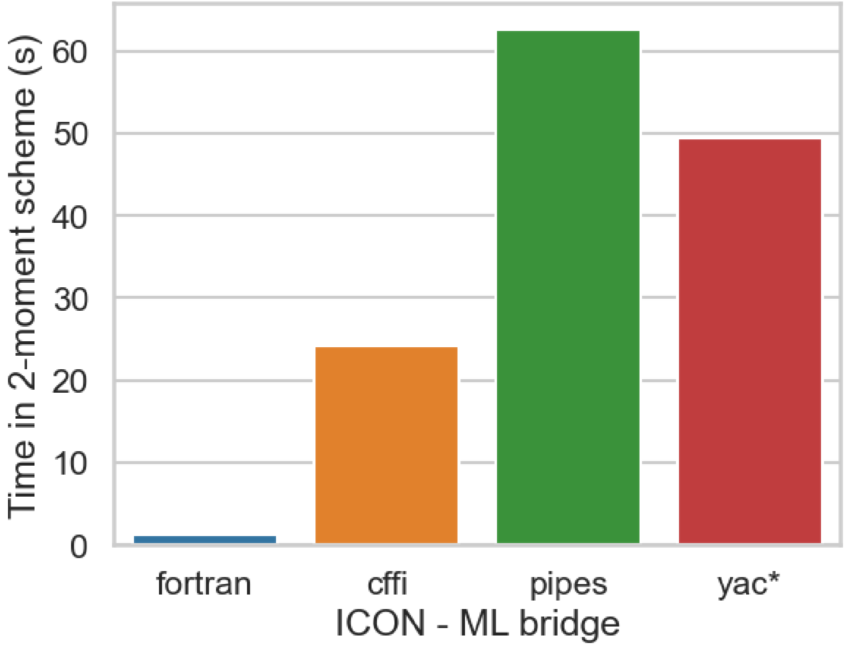


ML model by Shivani Sharma

Computational performance

Bubble scenario

- 880 horizontal cells, 70 vertical levels
- YAC bridge very much under development
- „Fortran“ – applies bulk moment scheme
- Overhead caused by
  - Application of ML model (< 1ms)
  - Data exchange



ICON – Python bridge

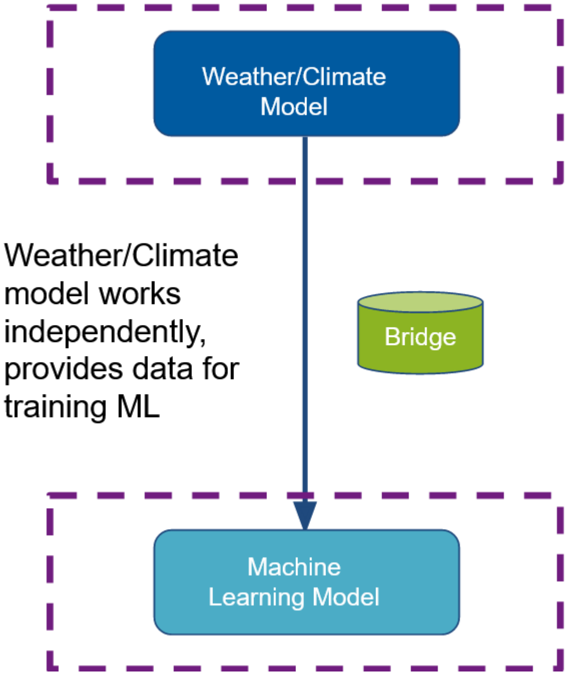
Qualitative comparison

flexible	Allow for iterative development of ML model	Embedded Python	YAC
ML-developer-friendly	Limit changes to ESM code	Embedded Python	YAC
performant	Runtime overhead should not limit development	Embedded Python	YAC
accelerated	Use ML specialised hardware (GPU) if possible	Embedded Python	YAC

Future use case: streaming training data

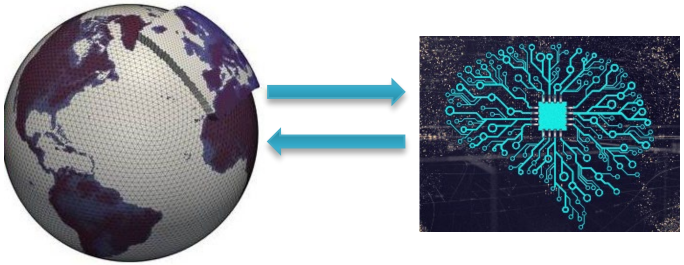
Attach any Python code via YAC

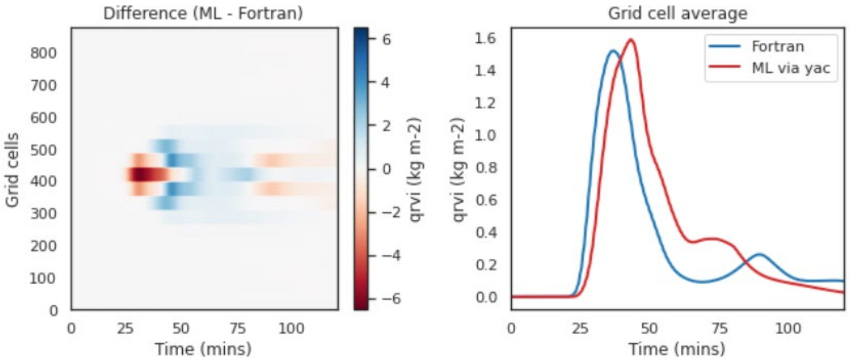
- ML models require large amounts of realistic training data
- Iterative training process
  - Receive current ICON time loop data
  - Advance ML model training by one „epoch“
- → Realistic training data
- → Reduced need for data storage
- → Reduced time for data loading in ML training



Summary

- ML models are becoming increasingly popular for Earth System modeling
- Important to move quickly to „online“ tests coupling ML model to ESM
- Demonstrated in the warm bubble scenario
  - Embedded Python
  - YAC
- Major challenge: bridging the communities of Machine Learning and Earth System Modeling





Thank you for your attention!

Questions?

- Contact: Caroline Arnold, arnold@dkrz.de
- We are hiring ☺

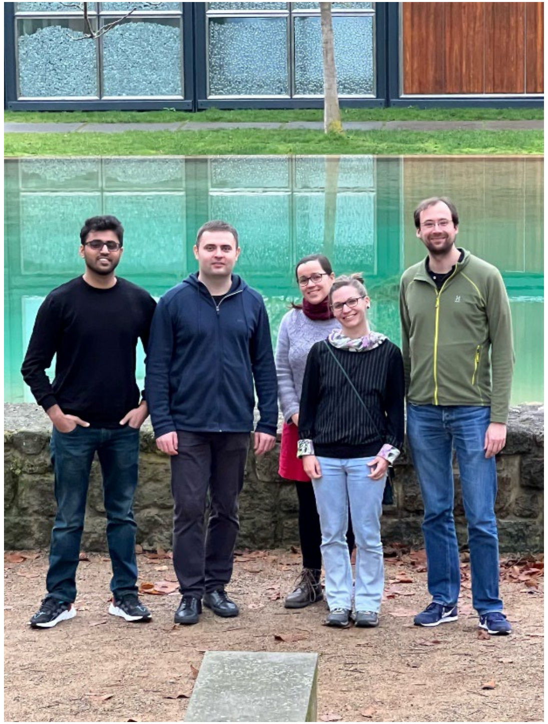
Acknowledgments

hereon
David Greenberg

MPI-M
Ann-Kristin Naumann
Sebastian Rast

DKRZ
Tobias Weigel
Moritz Hanke
Nils-Arne Dreier

Shivani Sharma



Summary

- ML models are becoming increasingly popular for Earth System modeling
- Important to move quickly to „online“ tests coupling ML model to ESM
- Demonstrated in the warm bubble scenario
  - Embedded Python
  - YAC
- Major challenge: bridging the communities of Machine Learning and Earth System Modeling