MAchinE Learning for Scalable meTeoROlogy and climate

The 2nd Dissemination Workshop of MAELSTROM

Ana Prieto Nemesio

www.maelstrom-eurohpc.eu

MAELSTROM

D4.8

The 2nd Dissemination Workshop of MAELSTROM

Author(s):	Ana Prieto Nemesio
Dissemination Level:	Public
Date:	24/11/2023
Version:	1.0
Contractual Delivery Date:	24/11/2023
Work Package/ Task:	WP4/ T4.8
Document Owner:	ECMWF
Contributors:	Peter Dueben, Jan Mirus
Status:	Final

MAELSTROM

Machine Learning for Scalable Meteorology and Climate

Research and Innovation Action (RIA)
H2020-JTI-EuroHPC-2019-1: Towards Extreme Scale Technologies and Applications

Project Coordinator: Dr Peter Dueben (ECMWF)
Project Start Date: 01/04/2021
Project Duration: 36 months

Published by the MAELSTROM Consortium

Contact:
ECMWF, Shinfield Park, Reading, RG2 9AX, United Kingdom
Peter.Dueben@ecmwf.int

The MAELSTROM project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955513. The JU receives support from the European Union's Horizon 2020 research and innovation programme and United Kingdom, Germany, Italy, Luxembourg, Switzerland, Norway

Contents

Figures

Tables

1 Executive Summary

The 2nd MAELSTROM Dissemination workshop took place on the 7th of November 2023. The celebration of this final Dissemination workshop included an overview of the progress of the MAELSTROM project as well as a series of talks across the three domains of machine learning, high-performance computing, and weather and climate sciences. The format of the workshop was hybrid, with the ability for participants to either join virtually from everywhere in the world or to attend in person at the ECMWF in Reading, UK. During this event, we had 36 in-person participants and over 200 online registrants listening to internal and external talks.

The first session of the dissemination workshop started with an introduction of the project MAELSTROM, followed by a series of presentations from MAELSTROM members focusing on specific aspects of the project, such as model performance benchmarking and an overview of the work done by Met Norway on the successful application of citizen observations and machine learning to improve their local forecast. The first session consisted of four talks:

- An introduction to MAELSTROM (Peter Düben, ECMWF)
- Performance Predictive Model for Deep Learning Model (Karthick Panner Selvam, University of Luxembourg)
- Experiences with W&C ML Apps on AMD Instinct GPUs (Stepan Nassyr, Jülich Supercomputing Center)
- A machine-learned weather forecast for Norway (Thomas Nipen, Norwegian Meteorological Institute)

In the second session, we heard from three different EuroHPC partner projects:

- SEA-Projects: towards a European heterogeneous system and SW architecture for Exascale and beyond (Hans-Christian Hoppe, Jülich Supercomputing Center)
- SparCity for Sparse Tensors: Study on Feature Extraction and Smart Tensor Generation (Tugba Torun, Koç University)
- Time-X: Some recent improvements of parallel-in-time algorithm (Daniel Ruprecht, Hamburg University of Technology)

The workshop concluded with two more sessions that featured a set of external talks that provided insight to other use cases within the field of machine learning applied to weather and climate science:

- Radiative transfer emulation: results so far and why we should move on to 3D (Peter Ukkonen, DMI)
- Ensemble member generation (Laure Raynaud, Météo-France)
- AIFS and AtmoRep (Simon Lang and Christian Lessig, ECMWF)
- Physics-Constrained Deep Learning for Downscaling and Emulation (Paula Harder, Fraunhofer Institute ITWM)
- Towards km-scale AI emulation for weather and climate applications (Karthik Kashinath, NVIDIA)

The talks on AIFS and AtmoRep were also of particular relevance for the MAELSTROM project as the two projects were supported by MAELSTROM scientists.

2 Introduction

2.1 About MAELSTROM

To develop Europe's computer architecture of the future, MAELSTROM will co-design bespoke compute system designs for optimal application performance and energy efficiency, a software framework to optimise usability and training efficiency for machine learning at scale, and large-scale machine learning applications for the domain of weather and climate science.

The MAELSTROM compute system designs will benchmark the applications across a range of computing systems regarding energy consumption, time-to-solution, numerical precision and solution accuracy. Customised compute systems will be designed that are optimised for application needs to strengthen Europe's high-performance computing portfolio and to pull recent hardware developments, driven by general machine learning applications, toward needs of weather and climate applications.

The MAELSTROM software framework will enable scientists to apply and compare machine learning tools and libraries efficiently across a wide range of computer systems. A user interface will link application developers with compute system designers, and automated benchmarking and error detection of machine learning solutions will be performed during the development phase. Tools will be published as open source.

The MAELSTROM machine learning applications will cover all important components of the workflow of weather and climate predictions including the processing of observations, the assimilation of observations to generate initial and reference conditions, model simulations, as well as post-processing of model data and the development of forecast products. For each application, benchmark datasets with up to 10 terabytes of data will be published online for training and machine learning tool-developments at the scale of the fastest supercomputers in the world. MAELSTROM machine learning solutions will serve as a blueprint for a wide range of machine learning applications on supercomputers in the future.

2.2 Scope of this deliverable

2.2.1 Objectives of this deliverable

This deliverable 4.8 is summarising the results of the second Dissemination Workshop:

- In line with the work carried out during the first Dissemination Workshop, the celebration of this event aimed to increase engagement and foster understanding within the scientific and technological communities about MAELSTROM's main pillars: weather and climate science, machine learning and High-performance computing.
- Relevant updates from the different MAELSTROM work packages about their latest development were shown to the wider community ahead of the project closure.

2.2.2 Work performed in this deliverable

Deliverable 4.8 provides a summary of the execution of the 2nd MAELSTROM Dissemination Workshop. In section 3, the work carried out ahead of the workshop is described: including its planning, preparation, and definition of the programme. Impressions from the workshop and the main conclusions drawn from the event are described in sections 4 and 5.

2.2.3 Deviations and countermeasures

No deviations occurred; hence no counter measures were requisite.

3 Planning, preparation and realisation of the workshop

3.1 Scheduling

The 2nd MAELSTROM Dissemination Workshop was held the 7th of November of 2023. In that same week, the General Assembly and the 2nd Bootcamp were also celebrated.

3.2 Infrastructure

The second dissemination workshop followed a hybrid format where participants could attend both on-line and in-person. All participants that attended the event were asked to register ahead to allow us to get an estimate of the size of the audience and adapt the protocol accordingly.

The registration page and mechanism, just like the technical support for the webinar infrastructure, was shared with the said ECMWF workshop; an efficient solution for MAELSTROM.
The workshop was announced on Twitter, on the MAELSTROM project website at https://www.maelstrom-eurohpc.eu/article?topic=dissemination-workshop-II and the events website at https://events.ecmwf.int/event/350/ (see Figure 1).

The workshop was broadcast from ECMWF's facilities in Reading UK. Microsoft Teams Meeting was the tool used to simplify virtual collaboration so that virtual participants could equally attend and interact during the event to increase the dissemination potential and to reduce the carbon budget of the event. All sessions were recorded, given previous consent of all participants. After each talk we had a Q&A round, where both online and on-site participants were able to ask questions.

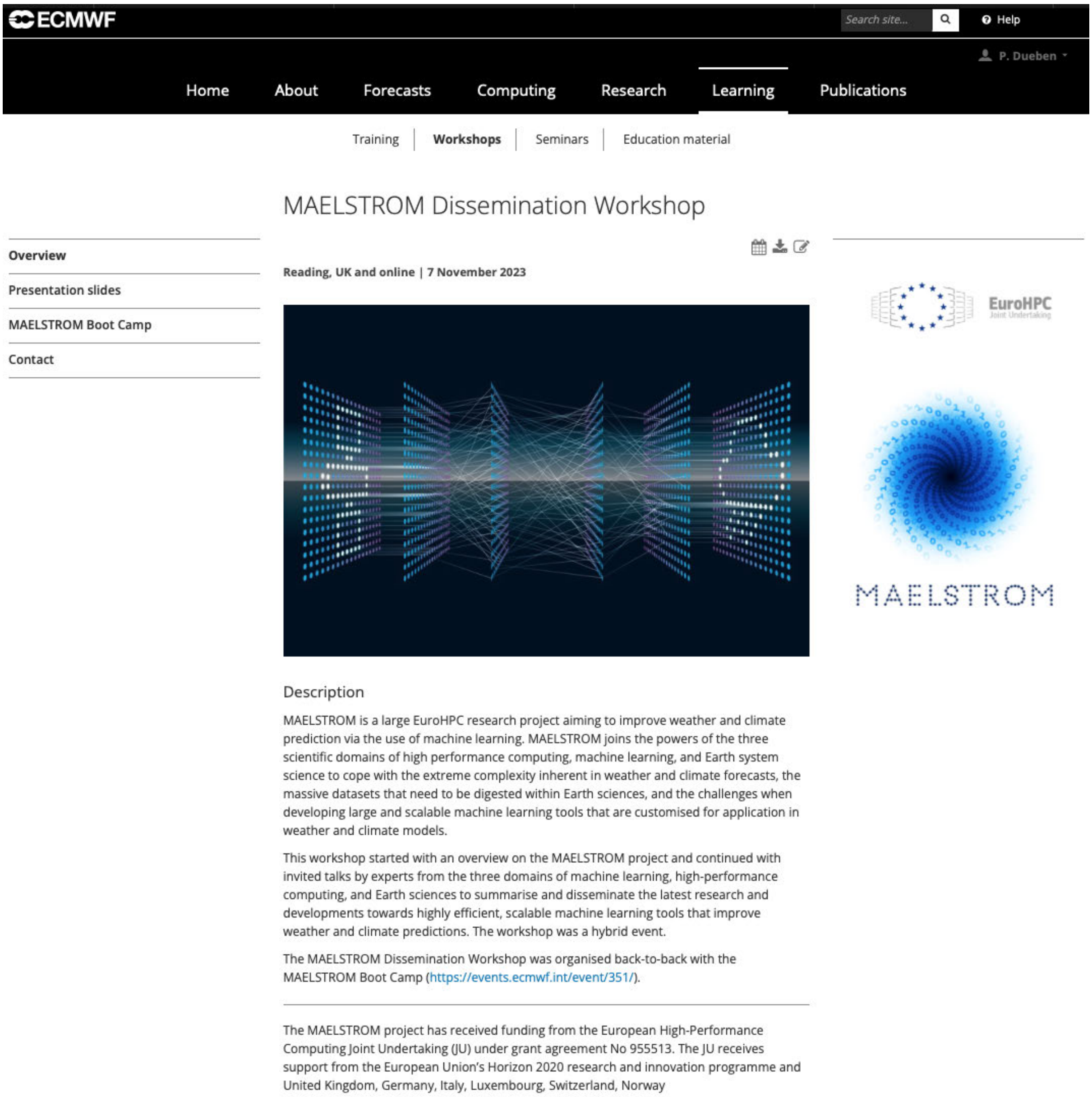


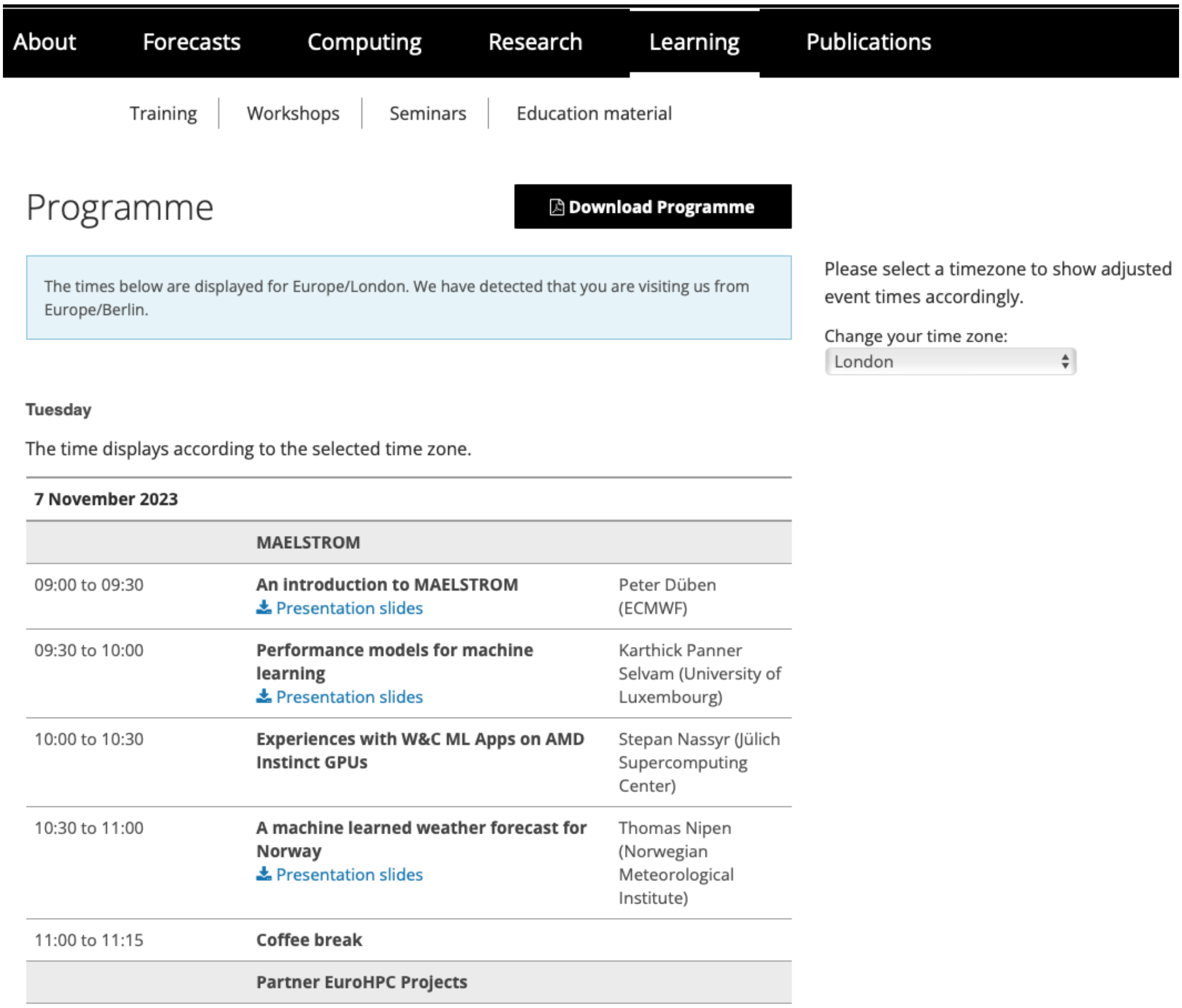
Figure 1: Screenshot of the website of the events that allowed users to register and to get information about the workshop.

3.3 Timetable

Part 1	Learn about MAELSTROM
09:00 → 09:30	**An introduction to MAELSTROM -** Peter Dueben (ECMWF)
09:30 → 10:00	**Performance Predictive Model for Deep Learning Model -** Karthick Panner Selvam

	(Uni Lu)
10:00 → 10:30	**Experiences with W&C ML Apps on AMD Instinct GPUs -** Stepan Nassyr (JSC)
10:30 → 11:00	**A machine learned weather forecast for Norway -** Thomas Nipen (Met Norway)
11:00 → 11:15	**Coffee Break**
Part 2	**EuroHPC Partner Projects Talks**
11:15 → 11:45	**SEA-Projects: towards a European heterogeneous system and SW architecture for Exascale and beyond** - Hans-Christian Hoppe (JSC)
11:45 → 12:15	**SparCity for Sparse Tensors: Study on Feature Extraction and Smart Tensor Generation** - Tugba Torun (Koç University)
12:15 → 12:45	**Some recent improvements of parallel-in-time algorithm -** Daniel Ruprecht (TU Hamburg)
12:45 → 14:00	**Lunch Break**
Part 3	**External Science Talks**
14:00 → 14:30	**Radiative transfer emulation: results so far and why we should move on to 3D** - Peter Ukkonen (DMI)
14:30 → 15:00	**Deep Learning for regional ensemble forecasting -** Laure Raynaud (Météo-France)
15:00 → 15:30	**AIFS** - Simon Lang (ECMWF)
15:30 → 16:00	**Coffee Break**
16:00 → 16:30	**AtmoRep: Large-Scale Representation Learning of Atmospheric Dynamics -** Christian Lessig (ECMWF)
16:30 → 17:00	**Physics-Constrained Deep Learning for Downscaling and Emulation -** Paula Harded (Fraunhofer Institute ITWM)
17:00 → 17:30	**Towards km-scale AI emulation for weather and climate applications** - Karthik Kashinath (NVIDIA)

The full programme is also available from the events page of the workshop (see Figure 2).


About | Forecasts | Computing | Research | Learning | Publications

Training | Workshops | Seminars | Education material

Programme

Download Programme

The times below are displayed for Europe/London. We have detected that you are visiting us from Europe/Berlin.

Please select a timezone to show adjusted event times accordingly.

Change your time zone:
London

Tuesday

The time displays according to the selected time zone.

7 November 2023

	MAELSTROM	
09:00 to 09:30	**An introduction to MAELSTROM** Presentation slides	Peter Düben (ECMWF)
09:30 to 10:00	**Performance models for machine learning** Presentation slides	Karthick Panner Selvam (University of Luxembourg)
10:00 to 10:30	**Experiences with W&C ML Apps on AMD Instinct GPUs**	Stepan Nassyr (Jülich Supercomputing Center)
10:30 to 11:00	**A machine learned weather forecast for Norway** Presentation slides	Thomas Nipen (Norwegian Meteorological Institute)
11:00 to 11:15	**Coffee break**	
	Partner EuroHPC Projects	

Figure 2: Screenshot of the interactive schedule from events page at https://ecmwfevents.com/i/e8f8e7dd-b114-4ef5-9f8c-393630425eae/public/agenda

3.4 Participation

A total of 40 participants on site and around 100 participants online were expected. On the day of the event, 36 in-person people participated on site and over 200 online registrants listened to internal and external talks in the general area of machine learning in weather and climate science. The workshop started and ended as scheduled.

4 Workshop impressions

Figure 3: Workshop assembled in ECMWF's council chamber.

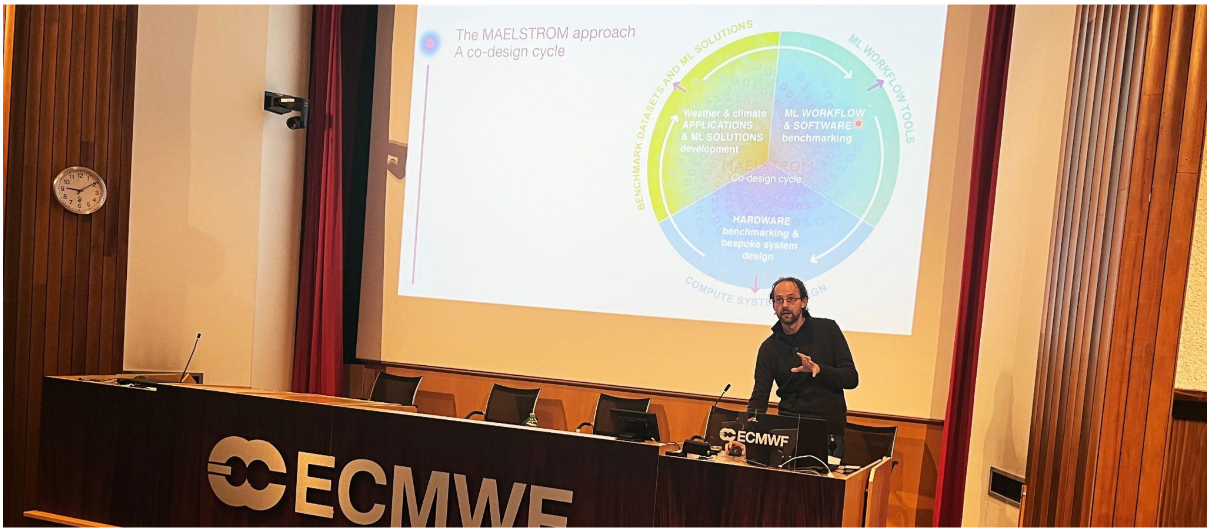


Figure 4: Introduction to MAELSTROM in ECMWF's lecture hall.

Figure 5: Networking was easy after the workshop.

5 Conclusion

The 2nd Dissemination workshop continued the work done during the 1st Dissemination Workshop, achieving this time a greater number of participants, with even some of them attending physically to the event. The workshop combined specific updates from MAELSTROM's project, talks from three different partner EURO HPC projects and a third session focused on a set of external talks from the machine learning community.

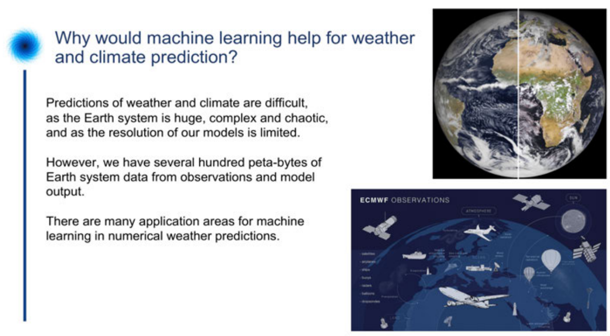
Participants could interact among each other and with the invited speakers either using Microsoft Teams Meeting during the Q&A sessions held at the end of each talk or at the different coffee breaks scheduled across the workshop.

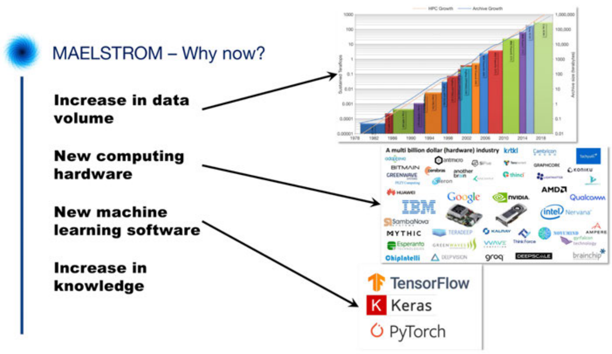
All presentation decks are available on the MAELSTROM[1] website and ECMWF's event page[2]. Furthermore, recordings of the event can be found here: https://drive.google.com/drive/folders/129J602-D1nNSY4i2bnrkIUXoOfDvYD8E?usp=sharing

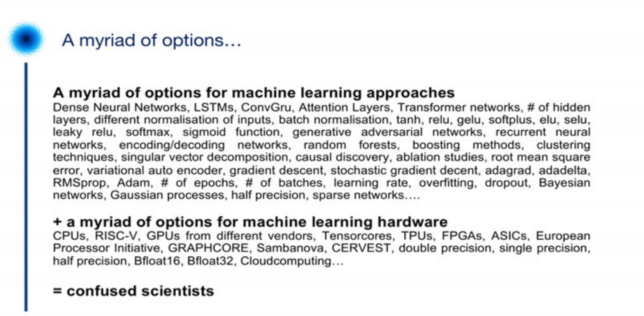
[1] https://www.maelstrom-eurohpc.eu/article?topic=dissemination-workshop-II
[2] https://ecmwfevents.com/i/e8f8e7dd-b114-4ef5-9f8c-393630425eae/public/agenda

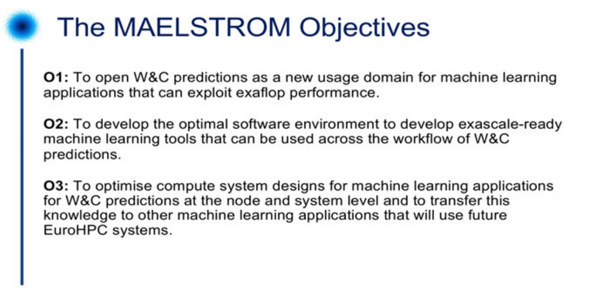
6 Annex: MAELSTROM presentation decks

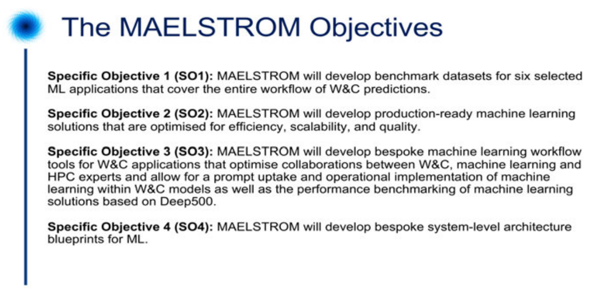
6.1 Introduction to MAELSTROM: Peter Dueben, ECMWF

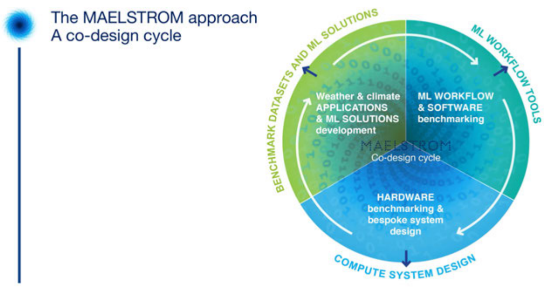


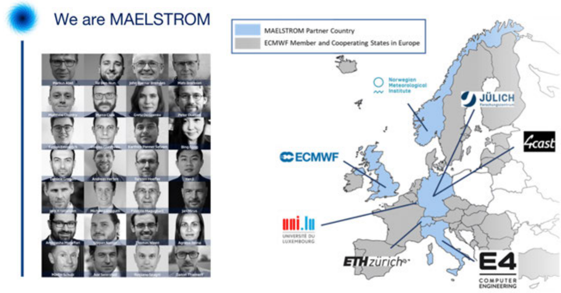




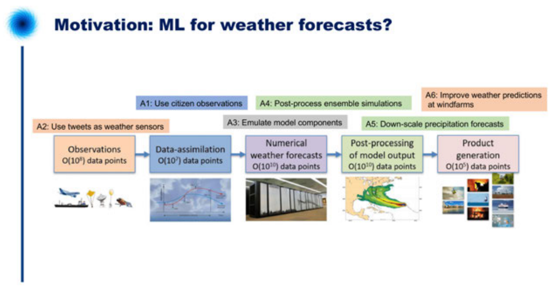


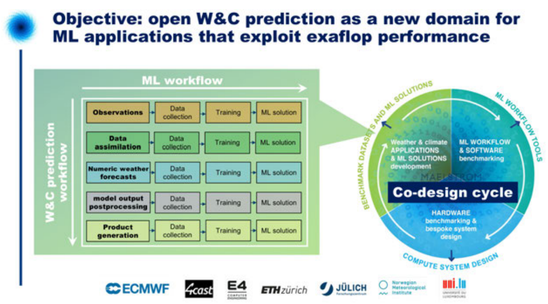


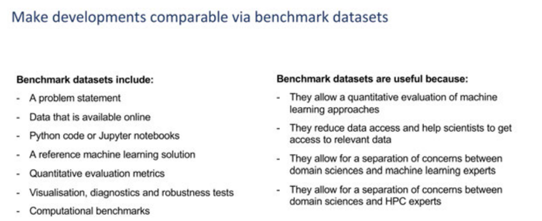

The MAELSTROM approach
A co-design cycle
BENCHMARK DATASETS AND ML SOLUTIONS
ML WORKFLOW TOOLS
Weather & climate APPLICATIONS & ML SOLUTIONS development
ML WORKFLOW & SOFTWARE benchmarking
MAELSTROM
Co-design cycle
HARDWARE benchmarking & bespoke system design
COMPUTE SYSTEM DESIGN

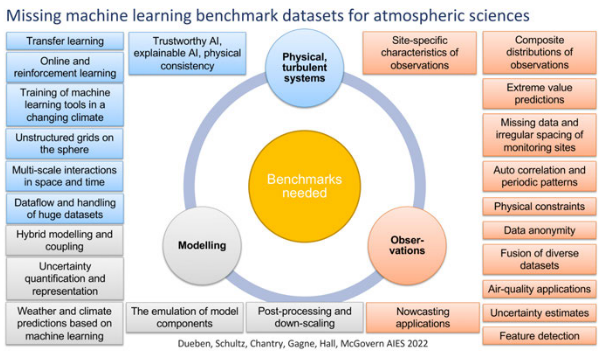

We are MAELSTROM
MAELSTROM Partner Country
ECMWF Member and Cooperating States in Europe
Norwegian Meteorological Institute
JÜLICH
4cast
ECMWF
uni.lu
ETH zürich
E4 COMPUTER ENGINEERING

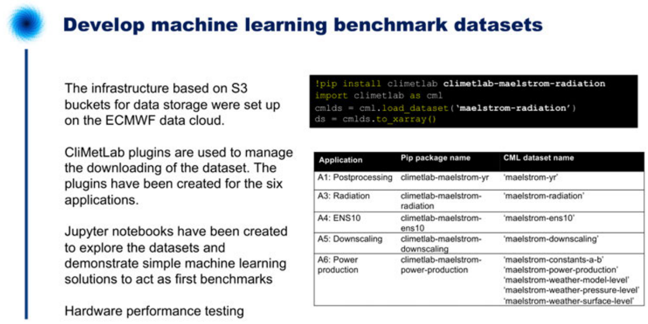

MAELSTROM
Empowering weather & climate forecast:
ML Apps & Datasets
ML Workflow Tools
Hardware Systems
ECMWF

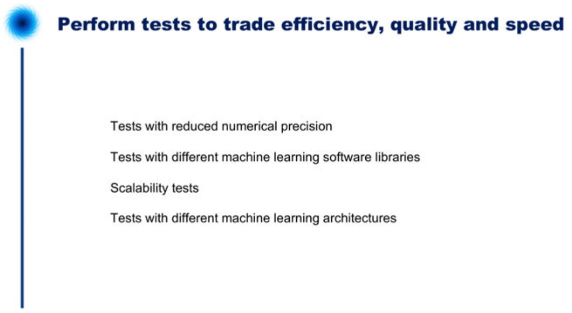

Motivation: ML for weather forecasts?
A1: Use citizen observations
A4: Post-process ensemble simulations
A6: Improve weather predictions at windfarms
A2: Use tweets as weather sensors
A3: Emulate model components
A5: Down-scale precipitation forecasts
Observations
Data-assimilation
Numerical weather forecasts
Post-processing of model output
Product generation

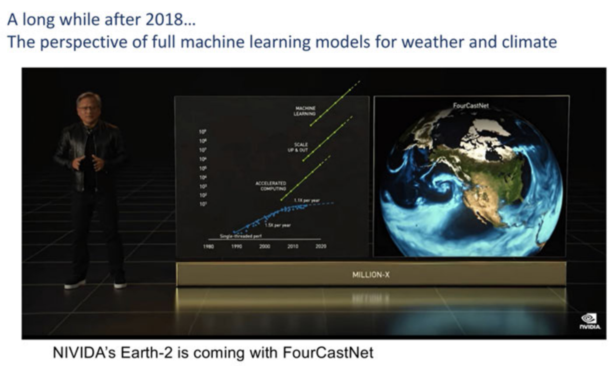

Objective: open W&C prediction as a new domain for ML applications that exploit exaflop performance
ML workflow
W&C prediction workflow
Observations
Data assimilation
Numeric weather forecasts
model output postprocessing
Product generation
Data collection
Training
ML solution
Co-design cycle
ECMWF
4cast
E4
ETH zürich
JÜLICH
Norwegian Meteorological Institute
uni.lu

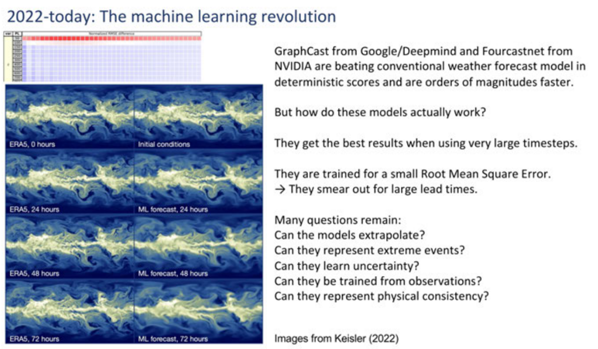

Make developments comparable via benchmark datasets
Benchmark datasets include:
- A problem statement
- Data that is available online
- Python code or Jupyter notebooks
- A reference machine learning solution
- Quantitative evaluation metrics
- Visualisation, diagnostics and robustness tests
- Computational benchmarks
Benchmark datasets are useful because:
- They allow a quantitative evaluation of machine learning approaches
- They reduce data access and help scientists to get access to relevant data
- They allow for a separation of concerns between domain sciences and machine learning experts
- They allow for a separation of concerns between domain sciences and HPC experts

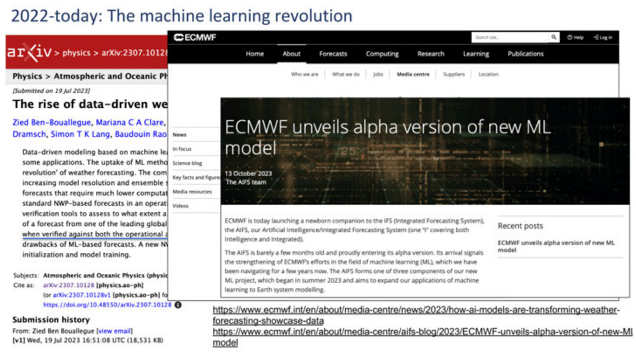

Missing machine learning benchmark datasets for atmospheric sciences
Transfer learning
Online and reinforcement learning
Training of machine learning tools in a changing climate
Unstructured grids on the sphere
Multi-scale interactions in space and time
Dataflow and handling of huge datasets
Hybrid modelling and coupling
Uncertainty quantification and representation
Weather and climate predictions based on machine learning
Trustworthy AI, explainable AI, physical consistency
Physical, turbulent systems
Benchmarks needed
Modelling
Obser-vations
The emulation of model components
Post-processing and down-scaling
Nowcasting applications
Site-specific characteristics of observations
Composite distributions of observations
Extreme value predictions
Missing data and irregular spacing of monitoring sites
Auto correlation and periodic patterns
Physical constraints
Data anonymity
Fusion of diverse datasets
Air-quality applications
Uncertainty estimates
Feature detection
Dueben, Schultz, Chantry, Gagne, Hall, McGovern AIES 2022

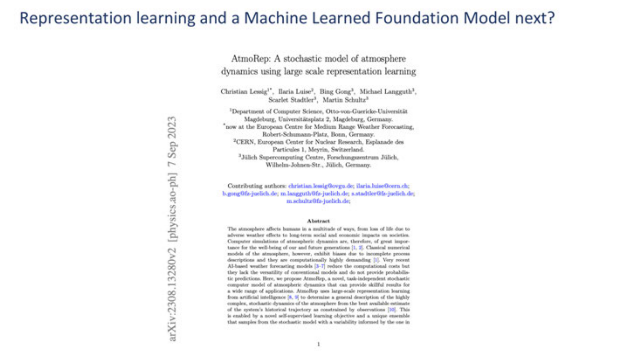

Develop machine learning benchmark datasets
The infrastructure based on S3 buckets for data storage were set up on the ECMWF data cloud.
CliMetLab plugins are used to manage the downloading of the dataset. The plugins have been created for the six applications.
Jupyter notebooks have been created to explore the datasets and demonstrate simple machine learning solutions to act as first benchmarks
Hardware performance testing
!pip install climetlab climetlab-maelstrom-radiation
import climetlab as cml
cmlds = cml.load_dataset('maelstrom-radiation')
ds = cmlds.to_xarray()
Application | Pip package name | CML dataset name
A1: Postprocessing | climetlab-maelstrom-yr | 'maelstrom-yr'
A3: Radiation | climetlab-maelstrom-radiation | 'maelstrom-radiation'
A4: ENS10 | climetlab-maelstrom-ens10 | 'maelstrom-ens10'
A5: Downscaling | climetlab-maelstrom-downscaling | 'maelstrom-downscaling'
A6: Power production | climetlab-maelstrom-power-production | 'maelstrom-constants-a-b' 'maelstrom-power-production' 'maelstrom-weather-model-level' 'maelstrom-weather-pressure-level' 'maelstrom-weather-surface-level'


Perform tests to trade efficiency, quality and speed
Tests with reduced numerical precision
Tests with different machine learning software libraries
Scalability tests
Tests with different machine learning architectures

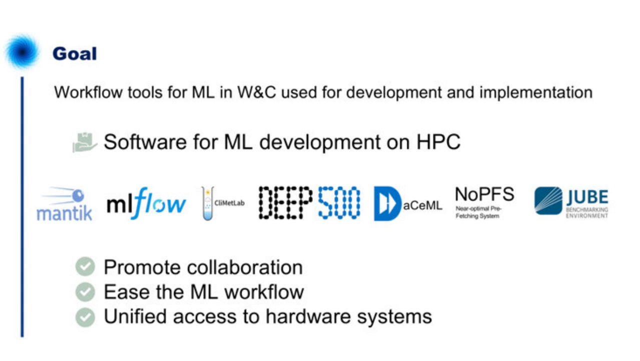

A long while after 2018…
The perspective of full machine learning models for weather and climate
FourCastNet
MILLION-X
NIVIDA's Earth-2 is coming with FourCastNet

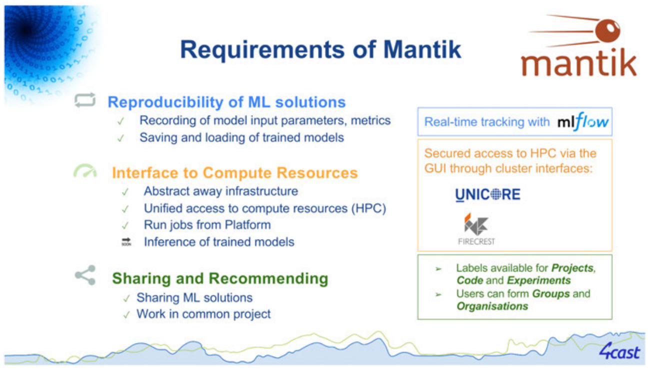

2022-today: The machine learning revolution
ERA5, 0 hours
Initial conditions
ERA5, 24 hours
ML forecast, 24 hours
ERA5, 48 hours
ML forecast, 48 hours
ERA5, 72 hours
ML forecast, 72 hours
GraphCast from Google/Deepmind and Fourcastnet from NVIDIA are beating conventional weather forecast model in deterministic scores and are orders of magnitudes faster.
But how do these models actually work?
They get the best results when using very large timesteps.
They are trained for a small Root Mean Square Error.
→ They smear out for large lead times.
Many questions remain:
Can the models extrapolate?
Can they represent extreme events?
Can they learn uncertainty?
Can they be trained from observations?
Can they represent physical consistency?
Images from Keisler (2022)

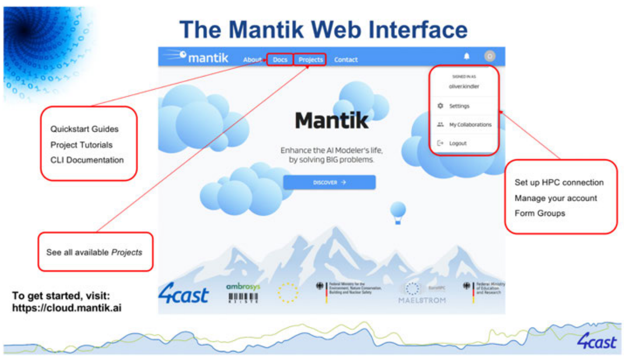

2022-today: The machine learning revolution
arXiv > physics > arXiv:2307.1012
Physics > Atmospheric and Oceanic Ph
[Submitted on 19 Jul 2023]
The rise of data-driven we
Zied Ben-Bouallegue, Mariana C A Clare, Dramsch, Simon T K Lang, Baudouin Rao
Submission history
From: Zied Ben Bouallegue [view email]
[v1] Wed, 19 Jul 2023 16:51:08 UTC (18,531 KB)
ECMWF
Home
About
Forecasts
Computing
Research
Learning
Publications
ECMWF unveils alpha version of new ML model
13 October 2023
The AIFS team
ECMWF is today launching a newborn companion to the IFS (Integrated Forecasting System), the AIFS, our Artificial Intelligence/Integrated Forecasting System (one "I" covering both Intelligence and Integrated).
The AIFS is barely a few months old and proudly entering its alpha version. Its arrival signals the strengthening of ECMWF's efforts in the field of machine learning (ML), which we have been navigating for a few years now. The AIFS forms one of three components of our new ML project, which began in summer 2023 and aims to expand our applications of machine learning to Earth system modelling.
Recent posts
ECMWF unveils alpha version of new ML model
https://www.ecmwf.int/en/about/media-centre/news/2023/how-ai-models-are-transforming-weather-forecasting-showcase-data
https://www.ecmwf.int/en/about/media-centre/aifs-blog/2023/ECMWF-unveils-alpha-version-of-new-ML-model

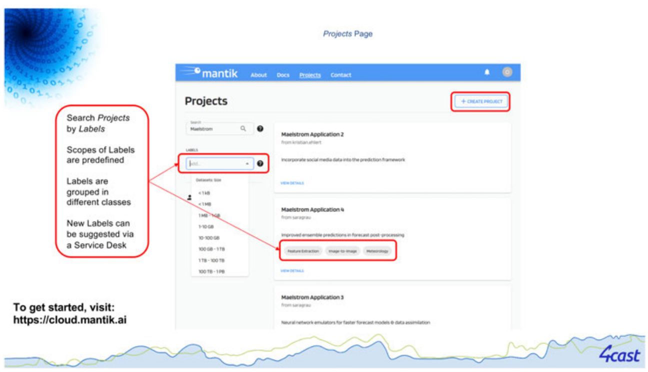

Representation learning and a Machine Learned Foundation Model next?
AtmoRep: A stochastic model of atmosphere dynamics using large scale representation learning
arXiv:2308.13280v2 [physics.ao-ph] 7 Sep 2023
Abstract

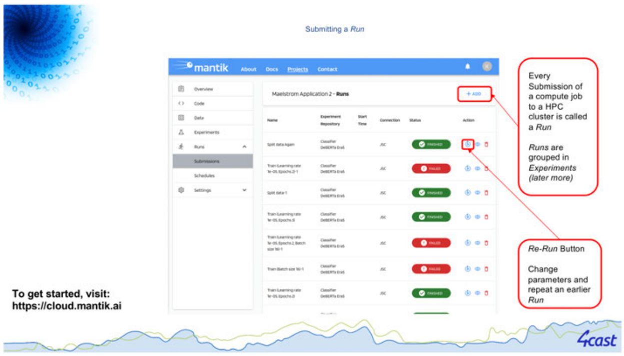

MAELSTROM
Empowering weather & climate forecast:
ML Apps & Datasets
ML Workflow Tools
Hardware Systems
ECMWF


Goal
Workflow tools for ML in W&C used for development and implementation
Software for ML development on HPC
mantik
mlflow
CliMetLab
DEEP 500
aCeML
NoPFS
Near-optimal Pre-Fetching System
JUBE
Promote collaboration
Ease the ML workflow
Unified access to hardware systems

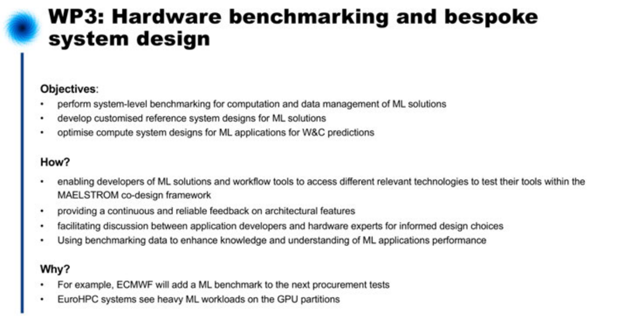

Requirements of Mantik
mantik
Reproducibility of ML solutions
Recording of model input parameters, metrics
Saving and loading of trained models
Real-time tracking with mlflow
Interface to Compute Resources
Abstract away infrastructure
Unified access to compute resources (HPC)
Run jobs from Platform
Inference of trained models
Secured access to HPC via the GUI through cluster interfaces:
UNICORE
FIRECREST
Sharing and Recommending
Sharing ML solutions
Work in common project
Labels available for Projects, Code and Experiments
Users can form Groups and Organisations
4cast

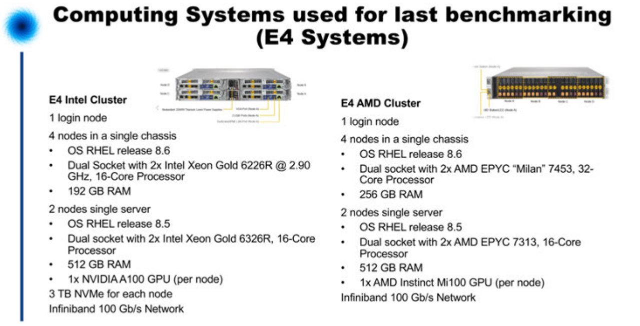

The Mantik Web Interface
Quickstart Guides
Project Tutorials
CLI Documentation
See all available Projects
Set up HPC connection
Manage your account
Form Groups
Mantik
Enhance the AI Modeler's life, by solving BIG problems.
To get started, visit:
https://cloud.mantik.ai
4cast

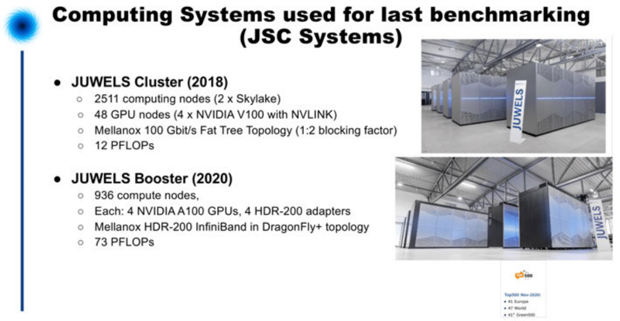

Projects Page
Projects
Search Projects by Labels
Scopes of Labels are predefined
Labels are grouped in different classes
New Labels can be suggested via a Service Desk
Maelstrom Application 2
Maelstrom Application 4
Maelstrom Application 3
To get started, visit:
https://cloud.mantik.ai
4cast

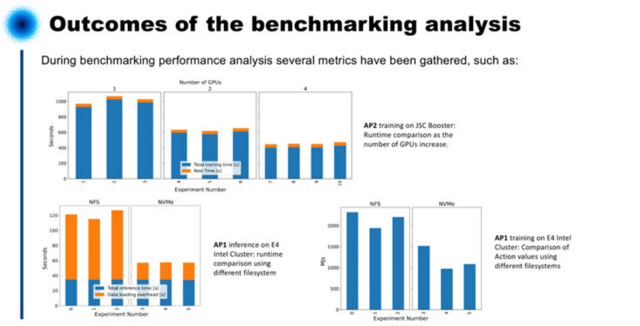

Submitting a Run
mantik
About
Docs
Projects
Contact
Overview
Code
Data
Experiments
Runs
Submissions
Schedules
Settings
Maelstrom Application 2 - Runs
+ ADD
Name
Experiment Repository
Start Time
Connection
Status
Action
Every Submission of a compute job to a HPC cluster is called a Run
Runs are grouped in Experiments (later more)
Re-Run Button
Change parameters and repeat an earlier Run
To get started, visit: https://cloud.mantik.ai
4cast

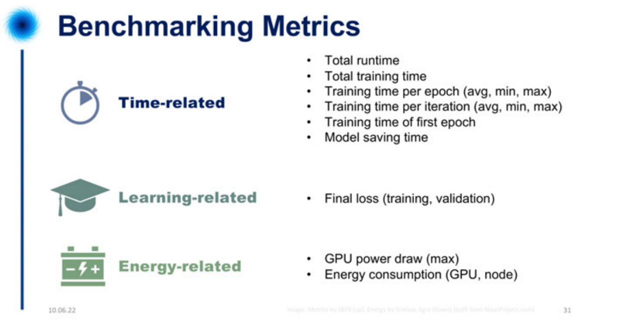

MAELSTROM
Empowering weather & climate forecast:
ML Apps & Datasets
ML Workflow Tools
Hardware Systems
"The MAELSTROM project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955513. The JU receives support from the European Union's Horizon 2020 research and innovation programme and United Kingdom, Germany, Italy, Luxembourg, Switzerland, Norway".
ECMWF

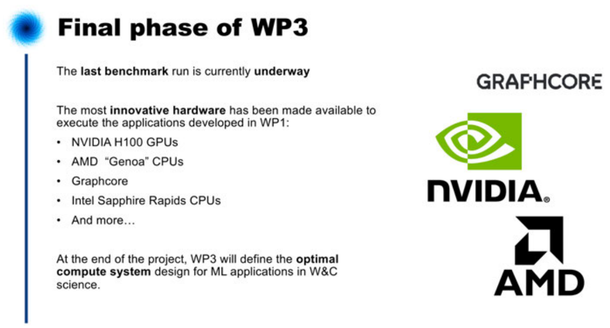

WP3: Hardware benchmarking and bespoke system design
Objectives:
• perform system-level benchmarking for computation and data management of ML solutions
• develop customised reference system designs for ML solutions
• optimise compute system designs for ML applications for W&C predictions
How?
• enabling developers of ML solutions and workflow tools to access different relevant technologies to test their tools within the MAELSTROM co-design framework
• providing a continuous and reliable feedback on architectural features
• facilitating discussion between application developers and hardware experts for informed design choices
• Using benchmarking data to enhance knowledge and understanding of ML applications performance
Why?
• For example, ECMWF will add a ML benchmark to the next procurement tests
• EuroHPC systems see heavy ML workloads on the GPU partitions


Computing Systems used for last benchmarking (E4 Systems)
E4 Intel Cluster
1 login node
4 nodes in a single chassis
• OS RHEL release 8.6
• Dual Socket with 2x Intel Xeon Gold 6226R @ 2.90 GHz, 16-Core Processor
• 192 GB RAM
2 nodes single server
• OS RHEL release 8.5
• Dual socket with 2x Intel Xeon Gold 6326R, 16-Core Processor
• 512 GB RAM
• 1x NVIDIA A100 GPU (per node)
3 TB NVMe for each node
Infiniband 100 Gb/s Network
E4 AMD Cluster
1 login node
4 nodes in a single chassis
• OS RHEL release 8.6
• Dual socket with 2x AMD EPYC "Milan" 7453, 32-Core Processor
• 256 GB RAM
2 nodes single server
• OS RHEL release 8.5
• Dual socket with 2x AMD EPYC 7313, 16-Core Processor
• 512 GB RAM
• 1x AMD Instinct Mi100 GPU (per node)
Infiniband 100 Gb/s Network


Computing Systems used for last benchmarking (JSC Systems)
• JUWELS Cluster (2018)
○ 2511 computing nodes (2 x Skylake)
○ 48 GPU nodes (4 x NVIDIA V100 with NVLINK)
○ Mellanox 100 Gbit/s Fat Tree Topology (1:2 blocking factor)
○ 12 PFLOPs
• JUWELS Booster (2020)
○ 936 compute nodes,
○ Each: 4 NVIDIA A100 GPUs, 4 HDR-200 adapters
○ Mellanox HDR-200 InfiniBand in DragonFly+ topology
○ 73 PFLOPs
JUWELS

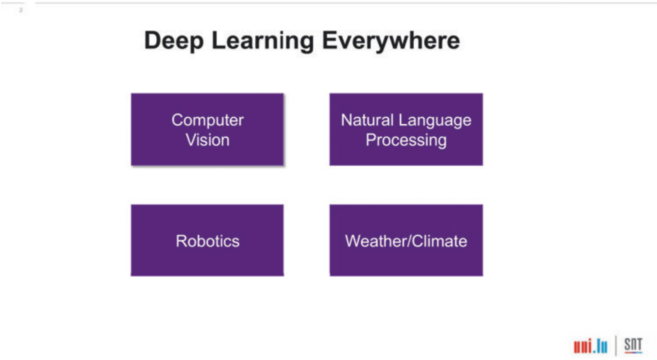

Outcomes of the benchmarking analysis
During benchmarking performance analysis several metrics have been gathered, such as:
Number of GPUs
Seconds
Total training time [s]
Rest Time [s]
Experiment Number
AP2 training on JSC Booster: Runtime comparison as the number of GPUs increase.
NFS
NVMe
Total inference time [s]
Data loading overhead [s]
AP1 inference on E4 Intel Cluster: runtime comparison using different filesystem
MJs
AP1 training on E4 Intel Cluster: Comparison of Action values using different filesystems

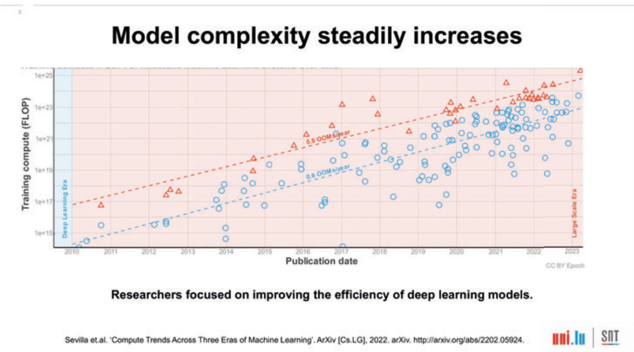


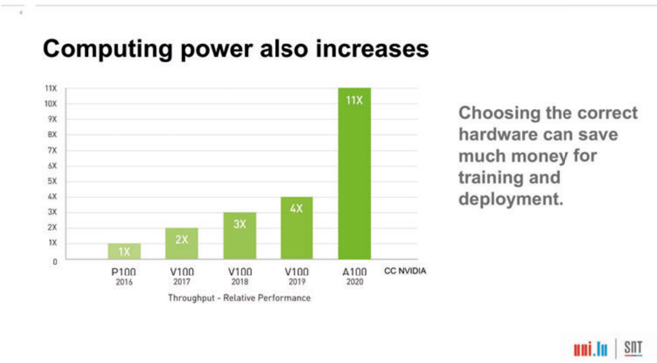


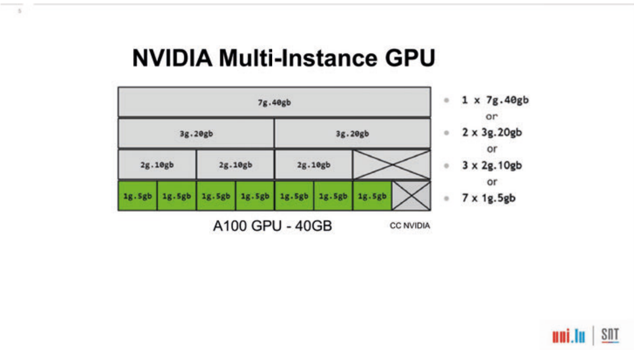
6.2 WP3: Performance Predictive Model for Deep Learning Model, Karthick Panner Selvam, University of Luxembourg

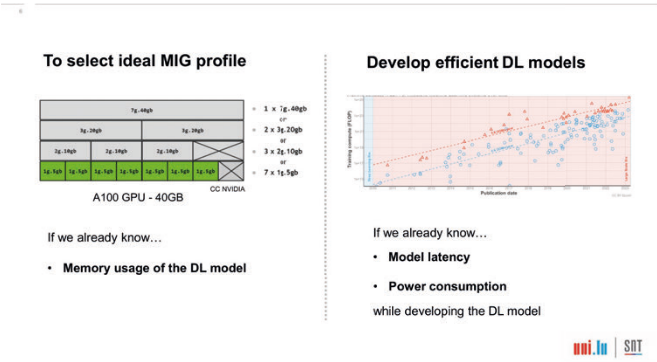

SnT
Performance Predictive Model for Deep Learning Models
Karthick Panner Selvam, Mats Brorsson
University of Luxembourg
karthick.pannerselvam@uni.lu
mats.brorsson@uni.lu
uni.lu
UNIVERSITY OF LUXEMBOURG

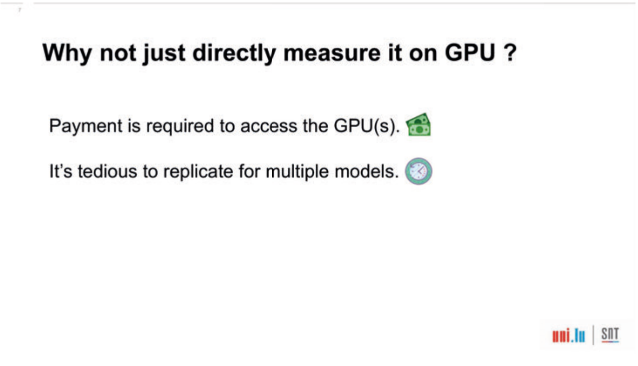

Deep Learning Everywhere
Computer Vision
Natural Language Processing
Robotics
Weather/Climate
uni.lu | SnT

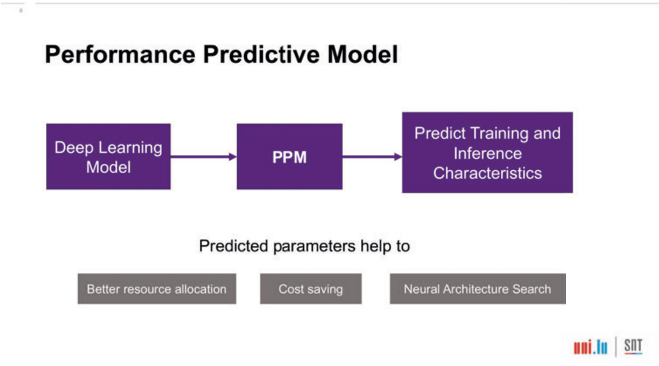

Model complexity steadily increases
Training compute (FLOP)
Publication date
Researchers focused on improving the efficiency of deep learning models.
Sevilla et.al. 'Compute Trends Across Three Eras of Machine Learning'. ArXiv [Cs.LG], 2022. arXiv. http://arxiv.org/abs/2202.05924.
uni.lu | SnT

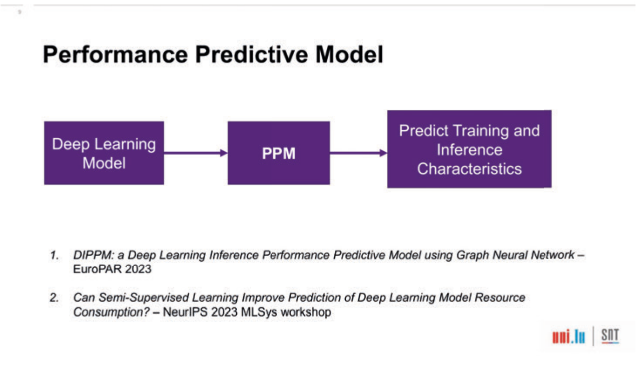

Computing power also increases
1X
2X
3X
4X
11X
P100 2016
V100 2017
V100 2018
V100 2019
A100 2020
CC NVIDIA
Throughput - Relative Performance
Choosing the correct hardware can save much money for training and deployment.
uni.lu | SnT

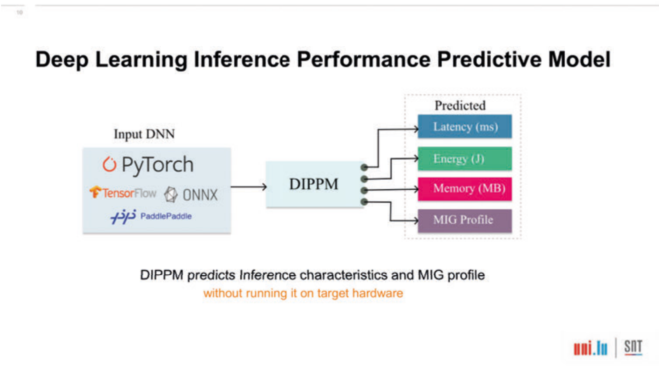

NVIDIA Multi-Instance GPU
7g.40gb
3g.20gb
3g.20gb
2g.10gb
2g.10gb
2g.10gb
1g.5gb
1 x 7g.40gb
or
2 x 3g.20gb
or
3 x 2g.10gb
or
7 x 1g.5gb
A100 GPU - 40GB
CC NVIDIA
uni.lu | SnT

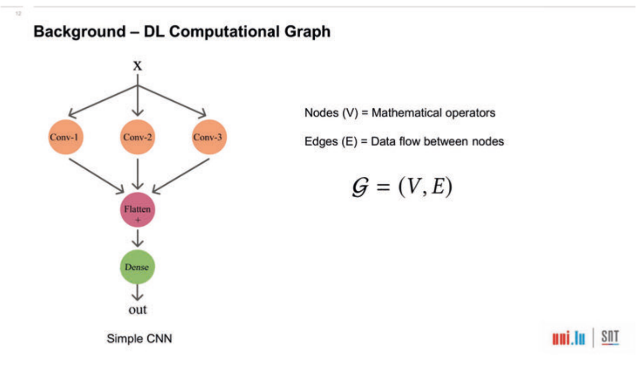

To select ideal MIG profile
A100 GPU - 40GB
CC NVIDIA
If we already know…
Memory usage of the DL model
Develop efficient DL models
If we already know…
Model latency
Power consumption
while developing the DL model
uni.lu | SnT

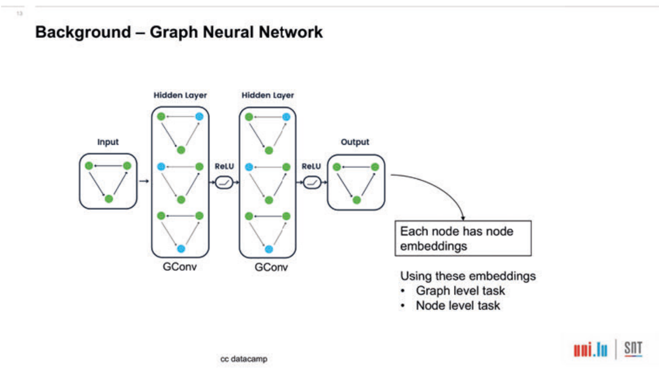

Why not just directly measure it on GPU ?
Payment is required to access the GPU(s).
It's tedious to replicate for multiple models.
uni.lu | SnT

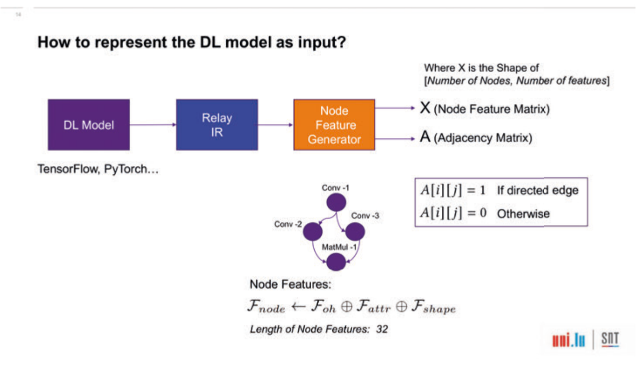

Performance Predictive Model
Deep Learning Model
PPM
Predict Training and Inference Characteristics
Predicted parameters help to
Better resource allocation
Cost saving
Neural Architecture Search
uni.lu | SnT

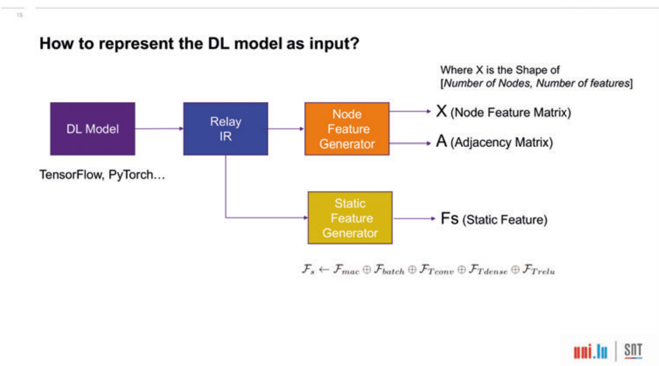

Performance Predictive Model
Deep Learning Model
PPM
Predict Training and Inference Characteristics
1. DIPPM: a Deep Learning Inference Performance Predictive Model using Graph Neural Network – EuroPAR 2023
2. Can Semi-Supervised Learning Improve Prediction of Deep Learning Model Resource Consumption? – NeurIPS 2023 MLSys workshop
uni.lu | SnT


Deep Learning Inference Performance Predictive Model
Input DNN
PyTorch
TensorFlow
ONNX
PaddlePaddle
DIPPM
Predicted
Latency (ms)
Energy (J)
Memory (MB)
MIG Profile
DIPPM predicts Inference characteristics and MIG profile
without running it on target hardware
uni.lu | SnT

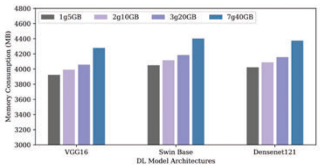

Background – DL Computational Graph
X
Conv-1
Conv-2
Conv-3
Flatten
Dense
out
Simple CNN
Nodes (V) = Mathematical operators
Edges (E) = Data flow between nodes
$\mathcal{G} = (V, E)$
uni.lu | SnT


Background – Graph Neural Network
Input
Hidden Layer
Hidden Layer
ReLU
ReLU
Output
GConv
GConv
Each node has node embeddings
Using these embeddings
• Graph level task
• Node level task
cc datacamp
uni.lu | SnT


How to represent the DL model as input?
Where X is the Shape of [Number of Nodes, Number of features]
DL Model
Relay IR
Node Feature Generator
X (Node Feature Matrix)
A (Adjacency Matrix)
TensorFlow, PyTorch...
$A[i][j] = 1$ If directed edge
$A[i][j] = 0$ Otherwise
Conv -1
Conv -2
Conv -3
MatMul -1
Node Features:
$\mathcal{F}_{node} \leftarrow \mathcal{F}_{oh} \oplus \mathcal{F}_{attr} \oplus \mathcal{F}_{shape}$
Length of Node Features: 32
uni.lu | SnT


How to represent the DL model as input?
Where X is the Shape of [Number of Nodes, Number of features]
DL Model
Relay IR
Node Feature Generator
X (Node Feature Matrix)
A (Adjacency Matrix)
TensorFlow, PyTorch...
Static Feature Generator
Fs (Static Feature)
$\mathcal{F}_s \leftarrow \mathcal{F}_{mac} \oplus \mathcal{F}_{batch} \oplus \mathcal{F}_{Tconv} \oplus \mathcal{F}_{Tdense} \oplus \mathcal{F}_{Trelu}$
uni.lu | SnT

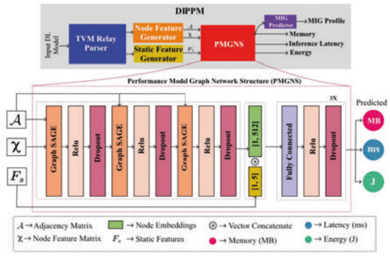
DIPPM: MIG Predictor



$$\mathrm{MIG}(\alpha) = \begin{cases} \text{1g.5gb}, & \text{if } 0gb < \alpha < \text{5gb} \\ \text{2g.10gb}, & \text{if 5gb} < \alpha < \text{10gb} \\ \text{3g.20gb}, & \text{if 10gb} < \alpha < \text{20gb} \\ \text{7g.40gb}, & \text{if 20gb} < \alpha < \text{40gb} \\ \text{None}, & \text{otherwise} \end{cases}$$

The memory consumption is always the highest when running on the 7g.40gb MIG profile. So, we claim that predicted memory will be an upper bound.

DIPPM: a Deep Learning Inference Performance Predictive Model using Graph Neural Network



DIPPM Architecture

DIPPM Dataset

- We used NVIDIA A100 GPU to collect the dataset. From 10 different model families, a total of **10508** graphs were collected.
- We used NVML and CUDA API to measure *Inference time*, *Memory*, and *Energy*.

Each graph contains

1. Node Features Matrix
2. Adjanceny Matrix
3. Target variable
4. Static Features

DIPPM: Results

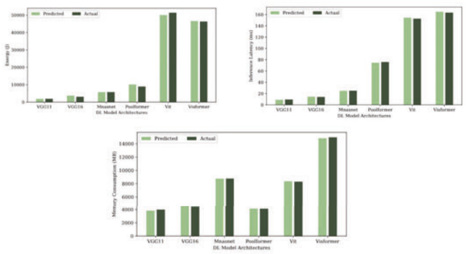
Model	Training Loss	Validation Loss
GAT	0.4966	0.3793
GCN	0.2122	0.1776
GIN	0.4880	0.3939
MLP	0.3714	0.3874
(Ours) GraphSAGE	0.1824	0.1587

In comparison with different GNN algorithms and MLP, we trained for ten epochs.

The results indicate that DIPPM with graphSAGE performs significantly better than other variants.

After 150 epochs, we achieved 1.9% MAPE on our test dataset.

DIPPM: Results



Results show that DIPPM predictions are close to the actual predictions.

DIPPM Usability

```
example.py M ×
example.py > ...
import dippm
import torchvision

model = torchvision.models.vgg16(pretrained=True)
model.eval()

#current dippm supports only A100 GPU
out  = dippm.predict(model, batch=8, input="3,244,244", device="A100")

print("====================================")
print("Predicted Memory {0} MB, Energy {1} J, Latency {2} ms, MIG {3}".format(*out))

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS 1

(multi) karthick@UNIDBP9NG3:~/work/dippm$
```

An example code demonstrating the utilization of DIPPM for performance prediction of a VGG16 DL model with a batch size of 8.

DIPPM: Summary

We developed a novel performance model to predict the *Inference characteristics* and *MIG profile* from a given input DL model from *various frameworks.*

uni.lu | SnT

TraPPM

Motivation:

Most prior studies, including DIPPM, utilized supervised techniques for performance prediction, *neglecting the vast pool of unlabelled DL model data*.

Our innovative approach, TraPPM, bridges this gap using a semi-supervised learning paradigm, enhancing prediction accuracy by harnessing unlabelled data.

Can Semi-Supervised Learning Improve Prediction of Deep Learning Model Resource Consumption? – NeurIPS 2023 MLSys workshop

uni.lu | SnT

TraPPM Dataset

- We used NVIDIA A100 and V100 to collect the dataset. From 11 different model families.
- We used NVML and CUDA API to measure *Training step time*, *Memory,* and *Power usage.*

Each graph contains

1. Node Features Matrix
2. Adjanceny Matrix
3. Target variable *(only for supervised)*
4. Static Features

$OneHot(Op_v)$	I_v	O_v	Mac_v	P_v	M_v
98	6	6	1	1	1

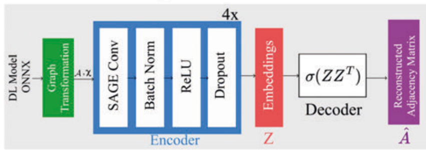
The graph's nodes are augmented with node features, each consisting of 113 elements.

Family	Unsupervised	Supervised	
		A100	V100
densenet	838	466	27
efficientnet	1370	566	44
mnasnet	7208	795	64
mobilenet	2449	1613	123
poolformer	601	377	36
resnet	1805	821	56
swin	787	421	36
vgg	6171	937	61
visformer	237	235	17
convnext	1530	439	27
vit	2057	866	52
Total	**25053**	**7536**	**543**

uni.lu | SnT

TraPPM: Unsupervised Learning

Graph Auto Encoder



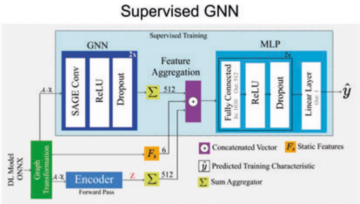
Training Graph Auto Encoder to minimize reconstruction loss of unlabelled DL model graphs

$$L_{\mathrm{BCE}} = -\log(\hat{A}(z, i_{\mathrm{pos}}, j_{\mathrm{pos}}) + \epsilon) - \log(1 - \hat{A}(z, i_{\mathrm{neg}}, j_{\mathrm{neg}}) + \epsilon)$$

uni.lu | SnT

TraPPM: Supervised Learning

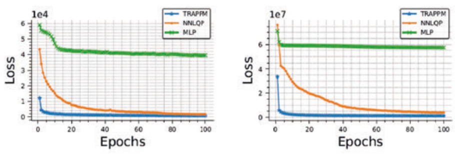
Supervised GNN



Training a GNN regressor using MSE loss to minimize the actual y vs. predicted y.

uni.lu | SnT

TraPPM: Results



(a) Step Time (ms) (b) Memory Usage (MB)

Epoch vs Loss plot *comparing the convergence rates of TraPPM, NNLQP, and MLP. TraPPM showcases rapid convergence due to its ability to leverage unsupervised learning from unlabeled data.*

uni.lu | SnT

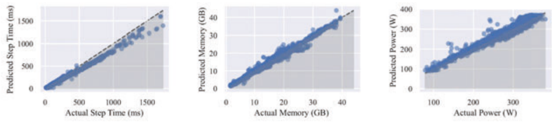
TraPPM: Results

Model	Memory Usage (MB) MAPE	Memory Usage (MB) RMSE	Step Time (ms) MAPE	Step Time (ms) RMSE
TraPPM	**4.92%**	**910.34**	**9.51%**	**23.23**
NNLQP	8.29%	1688.18	14.47%	37.02
MLP	85.01%	8045.68	134.07%	188.36
GBoost	16.10%	2971.52	16.98%	54.54

Average Performance Comparison of TraPPM with Baseline Models. The lower the value, the higher the accuracy.

uni.lu | SnT

TraPPM: Results

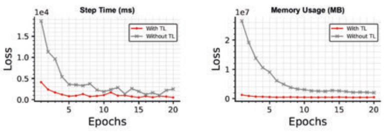


Comparison of actual values with predictions from TraPPM on the test set

uni.lu | SnT

TraPPM: Transfer Learning Results

Epoch vs. Loss plot demonstrating TraPPM's enhanced convergence through transfer learning.



Target variable	Metric	With TL	Without TL
Step Time	MAPE	**19.13%**	28.24%
	RMSE	**20.05 ms**	44.59 ms
Memory Usage	MAPE	**11.22%**	28.49%
	RMSE	**603.03 MB**	1176.90 MB

V100 Prediction results on the test dataset using with and without TL

uni.lu | SnT

TraPPM Usability

```
import trappm

trappm.predict("resnet101_32.onnx")
```

TraPPM Performance Prediction Report

GPU Metrics	A100 GPU Prediction
Train Memory	6633.54 Mb
Train Power	266.5 W
Train Step Time	58.52 ms
Inference Step Time	15.47 ms

Code: https://github.com/karthickai/trappm

uni.lu | SnT

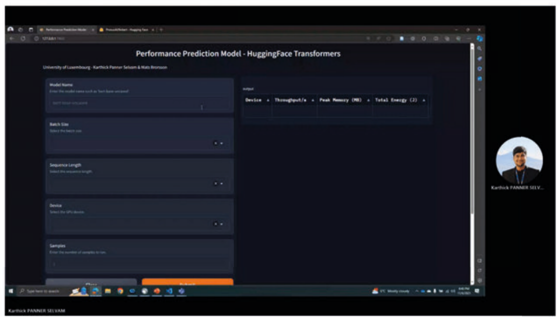
Summary

In DIPPM[1], we developed a novel performance model to predict the Inference characteristics and MIG profile.

In TraPPM[2], we utilized semi-supervised learning to use unlabeled data to enhance performance accuracy.

1. *DIPPM: a Deep Learning Inference Performance Predictive Model using Graph Neural Network* – EuroPAR 2023
2. *Can Semi-Supervised Learning Improve Prediction of Deep Learning Model Resource Consumption?* – NeurIPS 2023 MLSys workshop

uni.lu | SnT



Performance Transformer – Ongoing research

uni.lu | SnT

6.3 WP3: Experiences with W&C ML Apps on AMD Instinct GPUs, Stepan Nassyr (JSC)

MAELSTORM ON AMD

MI250x vs V100 vs A100 vs H100

November 7, 2023 | Stepan Nassyr | JSC

Member of the Helmholtz Association

JÜLICH Forschungszentrum

V100 nodes (JUWELS Cluster):

CPU	2 × Intel Xeon Gold 6148, (2×20 cores@2.4 GHz)
GPU	4 × NVIDIA V100 GPU, 16 GB HBM
Memory	196GiB DDR4-2666
NIC	2 × Mellanox EDR InfiniBand ConnectX 4

A100 nodes (JUWELS Booster):

CPU	2 × AMD EPYC 7402 (2×24 cores@2.8Ghz)
GPU	4 × NVIDIA A100 GPU, 40 GB HBM
Memory	512GiB DDR4-3200
NIC	4 × Mellanox HDR InfiniBand ConnectX 6

H100 node (JURECA DC Eval.):

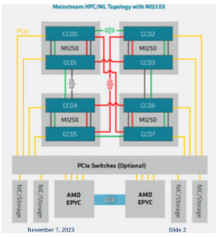
CPU	2 × Intel Xeon Platinum Sapphire Rapid 8452Y (2×36 cores@2.0Ghz)
GPU	4 × NVIDIA H100 PCIe GPU, 80 GB HBM
Memory	512GiB DDR5-4800
NIC	1 × BlueField-2 ConnectX-6 DPU

Member of the Helmholtz Association November 7, 2023 Slide 1

JÜLICH Forschungszentrum

AMD GPU nodes @ JSC:

CPU	2 × AMD EPYC 7443 (2×24 cores@2.85Ghz)
GPU	4 × AMD MI250 GPUs, 8 GCDs with 64GB HBM each
Memory	512GiB DDR4-3200
NIC	1 × Mellanox HDR InfiniBand ConnectX 6

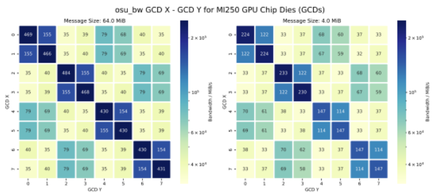


Member of the Helmholtz Association November 7, 2023 Slide 2

JÜLICH Forschungszentrum

AMD GPU nodes @ JSC:

- 2 GCDs per card
- assymmetric chip-to-chip bandwidth



Member of the Helmholtz Association November 7, 2023 Slide 3

JÜLICH Forschungszentrum

Theoreticals Mi250 vs A100:

Metric	MI250	A100
FP64	45.3 TFLOP/s	19.5 TFLOP/s
FP32	90.5 TFLOP/s	19.5 TFLOP/s
TF32	-	156|312 (dense|sparse) TFLOP/s
FP16	362 TFLOP/s	312|624 (dense|sparse) TFLOP/s

Member of the Helmholtz Association November 7, 2023 Slide 4

JÜLICH Forschungszentrum

Software:

- MI250x: container - tensorflow_rocm5.7-tf2.13-dev.sif
- V100: JWC EasyBuild modules
- A100: JWB EasyBuild modules
- H100: container - tensorflow_23.10-tf2-py3.sif

Member of the Helmholtz Association November 7, 2023 Slide 5

JÜLICH Forschungszentrum

Containers + VENV tricky:

- MI250x
  - run container
  - create venv without pip
  - install into container with prefix
- H100
  - run container
  - venv doesn't work at all
  - install into directory with prefix
  - set PYTHONPATH manually
- Guide/sourceable env scripts for D3.7 WIP

Member of the Helmholtz Association November 7, 2023 Slide 6

JÜLICH Forschungszentrum

Measuring Energy:

- AMD: energy counter

```
~]$ rocm-smi --showenergycounter --csv
```

- NVIDIA: Measure power and integrate

```
~]$ nvidia-smi --query-gpu=index,timestamp,name,power.draw --format=csv --loop-ms=5
```

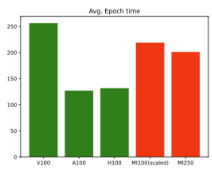
```
with GetPower() as measured_scope:
    print('Measuring Energy during main() call')
    try:
        main(args)
    except Exception as exc:
        print(f"Errors during training: {exc}")
print("Energy data:")
f = open(f"EnergyFile-NVDA-{args.id}", 'a')
print(measured_scope.df.groupby('index').get_group(0))
f.write(str(measured_scope.df.groupby('index').get_group(0)))
print("Energy-per-GPU-list:")
print(measured_scope.energy())
f.write(str(measured_scope.energy()))
```

Member of the Helmholtz Association November 7, 2023 Slide 7

JÜLICH Forschungszentrum

AP5 Results:

- 1 x MI100 (scaled from D3.6) : 218s/epoch
- 0.5 x MI250 (1 GCD): 198-212s/epoch
- 1x V100: 255-260s/epoch
- 1x A100: 125-130s/epoch
- 1x H100: 128-145s/epoch (unoptimized?)

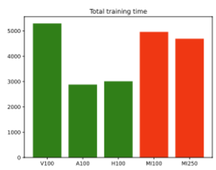


Member of the Helmholtz Association November 7, 2023 Slide 8

JÜLICH Forschungszentrum

AP5 Results (total training time):

- 1 x MI100 (from D3.6) : 4962s
- 0.5 x MI250 (1 GCD) (scaled): 4691s
- 1x V100 (from D3.6): 5294s
- 1x A100 (from D3.6): 2882s
- 1x H100 (scaled): 3011s

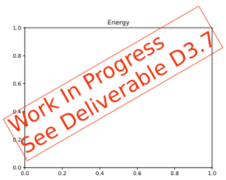


Member of the Helmholtz Association November 7, 2023 Slide 9

JÜLICH Forschungszentrum

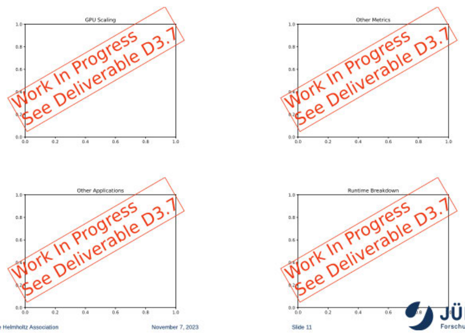
AP5 Results (Energy):

- MI100 : no GPU measurement
- MI250 : 3723Wh ??
- V100 (from D3.6): 131Wh
- A100 (from D3.6): 83Wh
- H100: crash



Member of the Helmholtz Association November 7, 2023 Slide 10

JÜLICH Forschungszentrum



Member of the Helmholtz Association November 7, 2023 Slide 11

JÜLICH Forschungszentrum

WIP - ironing out kinks

- container issues (both AMD+NVDA)
- containers mentioned before work well
- 0.5 × Mi250 results promising
- H100 appears underutilized - worth investigating
- Expect more interesting results in D3.7!

Member of the Helmholtz Association November 7, 2023 Slide 12

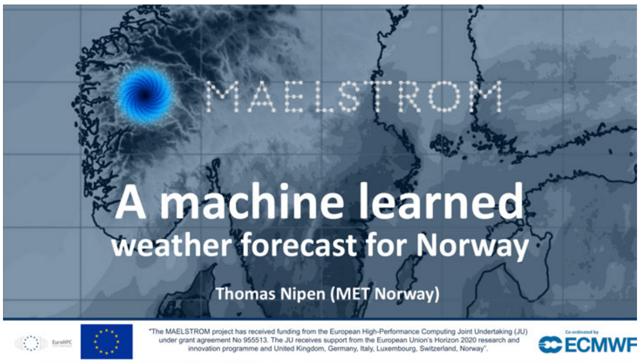
JÜLICH Forschungszentrum

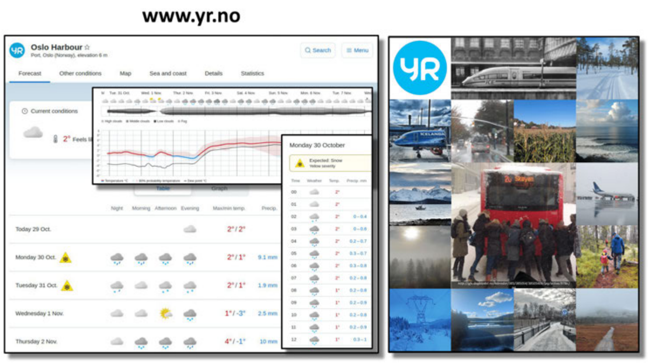
Thank you for your attention!

Member of the Helmholtz Association November 7, 2023 Slide 13

JÜLICH Forschungszentrum

6.4 WP1: A machine learned weather forecast for Norway, Thomas Nipen, MET Norway)

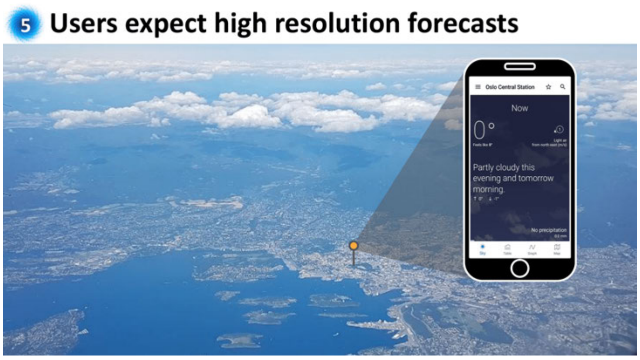


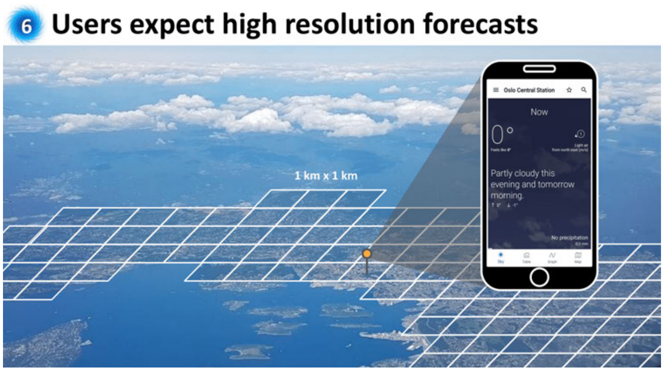


Part 1: Application

Part 2: ML solution

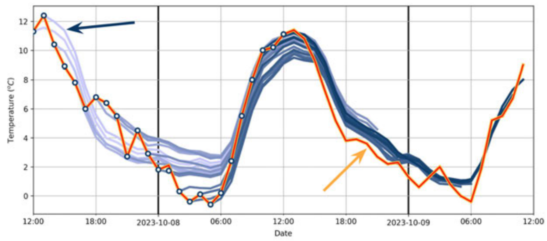
Part 3: Evaluation





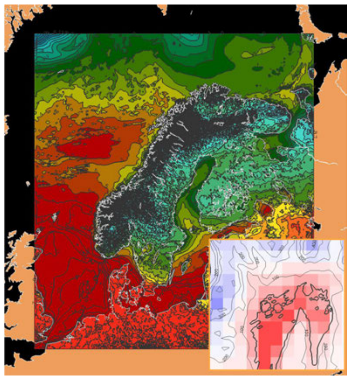
7 Users expect up-to-date forecasts

- New forecasts issued every hour as new observations become available



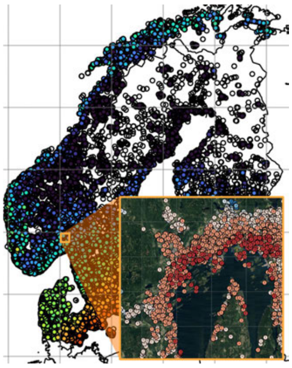
8 Predictors

- High resolution NWP ensemble (2.5 km)
- Hourly output for 59 hours
- Predictors:
  - 2m temperature (ensemble control)
  - 2m temperature (ensemble 10%)
  - 2 temperature (ensemble 90%)
  - 1h precipitation accumulation
  - Cloud cover
  - 10m wind (x-component)
  - 10m wind (y-component)
- Metadata variables:
  - Model altitude
  - Model land area fraction
  - "Real" altitude (1x1 km)
  - "Real" land area fraction (1x1 km)
  - Model x-coordinate
  - Model y-coordinate



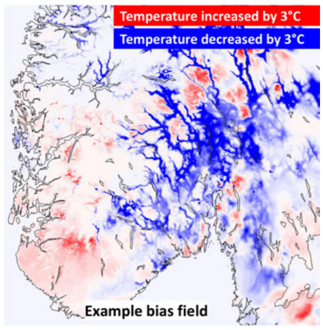
9 Target data

- Challenge to assemble an accurate target at high resolution
- Conventional observation networks are too sparse (at least in the Nordics)
- Citizen observations are an emerging data source (50-100x increase compared to SYNOP network)
- Target field based on:
  - Citizen observations
  - Early lead times (3-9h) from NWP
  - Combined using optimal interpolation (OI)



10 Gridded truths as input predictors

- Target fields for the 24h leading up to prediction also used as input predictors
- Allows us to keep forecasts up to date with recent observations
- NWP bias (target - NWP) used as predictor



11 Prediction problem

Input data (6 terabytes)

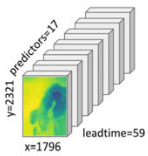
- 1x1 km downscaled NWP and recent biases
- 59 x 2321 x 1796 x 17
- 700 samples (2 years)

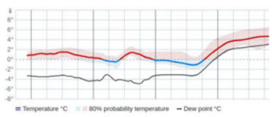
Output

- 1x1 km temperature forecasts
- 59 x 2321 x 1796 x 3 (10, 50, 90% quantile levels)

Target data

- 1x1 km gridded truth
- 59 x 2321 x 1796

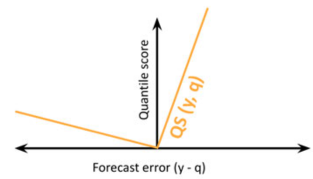




12 Loss function

- Quantile scoring function is used to evaluate quantile forecasts (10, 50, 90%)

Example for quantile level = 0.9



Quantile score

$$QS_\tau(y,q) = \begin{cases} (y-q)(\tau-1)) & y \le q \\ (y-q)\tau & y > q \end{cases}$$

τ : quantile level
y : observation
q : quantile

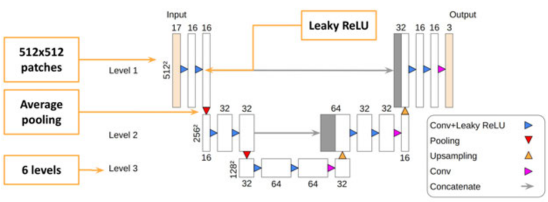
Part 1: Application

Part 2: ML solution

Part 3: Evaluation

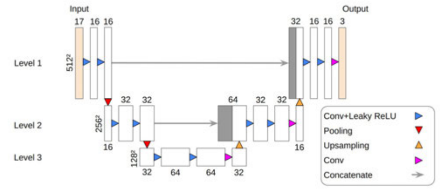
14 U-Net

- 2D U-Net, all leadtimes trained together (leadtime added as a predictor)
- 1,314,019 trainable parameters



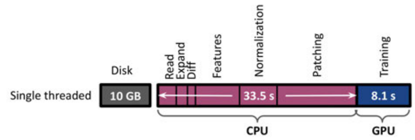
15 U-Net

- 2D U-Net, all leadtimes trained together (leadtime added as a predictor)
- 1,314,019 trainable parameters
- Trained on 4 NVIDIA A-100 GPUs, 2x24 cores AMD EPYC 7402, 512GB RAM
- Extensive optimization of processing performance and memory footprint



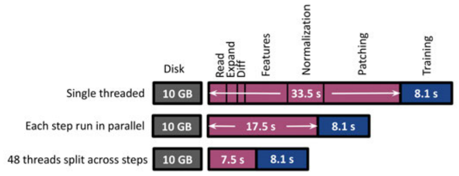
16 Optimizing the data loader

- We needed a data loader that:
  - Streames data from disk (6TB too large for memory)
  - Doesn't cause an I/O bottleneck
  - Can read data as we have them stored on our systems (i.e. reusable in other applications)
  - Allows loading options to be easily changed



17 Optimizing the data loader

- We needed a data loader that:
  - Streames data from disk (6TB too large for memory)
  - Doesn't cause an I/O bottleneck
  - Can read data as we have them stored on our systems (i.e. reusable in other applications)
  - Allows loading options to be easily changed



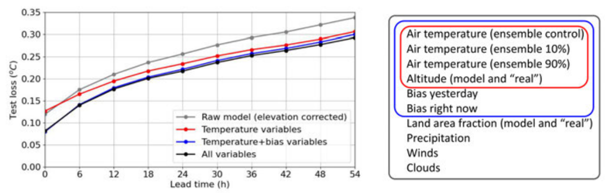
Part 1: Application

Part 2: ML solution

Part 3: Evaluation

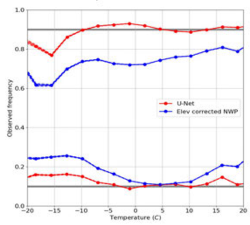
19 Verification

- 1 year training period, 1 year testing period
- Bias variables are important contributors to overall skill of forecast
- Precip/winds/clouds also have a (small) positive effect



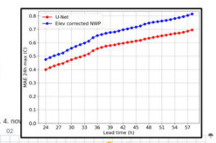
20 Verification

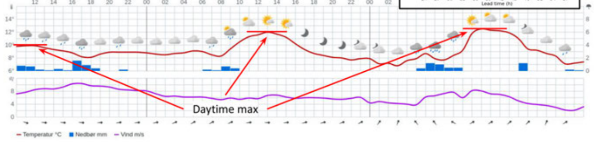
- 1 year training period, 1 year testing period
- Bias variables are important contributors to overall skill of forecast
- Precip/winds/clouds also have a (small) positive effect
- 10 and 90% quantiles are much more reliable



21 Verification

- Other properties not captured by the loss function:
  - Daily min/max
  - Sharp temporal changes
  - Spatial consistency (users comparing different locations)
- So far, these metrics also look promising





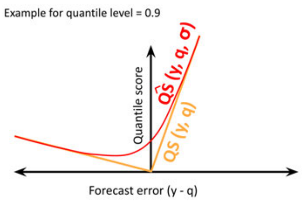
22 Summary

- The MAELSTROM project has contributed to:
  - Development of an ML solution for forecasting temperature suitable for the general public
  - Optimization of the training pipeline by exploiting the available hardware
  - Development of a high-resolution benchmarking dataset for testing new ML methods
- Links:
  - Forecast site: www.yr.no
  - Data access via climetlab: https://github.com/metno/maelstrom-yr
  - Jupyter notebooks: https://gitlab.jsc.fz-juelich.de/esde/training/maelstrom_bootcamp (AP1)
  - Contact: Thomas Nipen (thomasn@met.no)

This presentation reflects the views only of the author, and the European High-Performance Computing Joint Undertaking or Commission cannot be held responsible for any use which may be made of the information contained therein.

24 Loss function

- Quantile scoring function is used to evaluate quantile forecasts (10, 50, 90%)
- Convolve the forecast error with the target uncertainty
- Developed a computationally efficient approximation



Quantile score

$$QS_\tau(y,q) = \begin{cases} (y-q)(\tau-1)) & y \le q \\ (y-q)\tau & y > q \end{cases}$$

τ : quantile level
y : observation
q : quantile

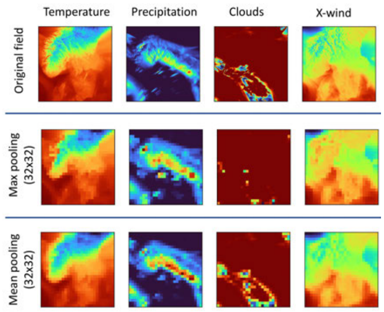
Quantile score (uncertain target)

$$QS_\tau(y,q,\sigma) = \int_{-\infty}^{+\infty} QS_\tau(y-x,q)\phi\left(\frac{x}{\sigma}\right)dx$$

$$\hat{QS}_\tau(y,q,\sigma) = QS_\tau(y,q) + 0.4\sigma e^{\frac{1.4|y-q|}{\sigma}}$$

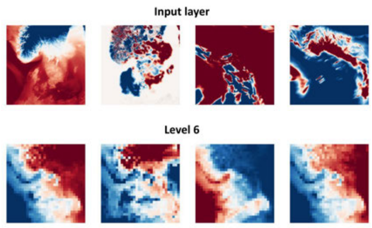
25 Average pooling

- Improvements also found in U-Net in MAELSTROM application 5



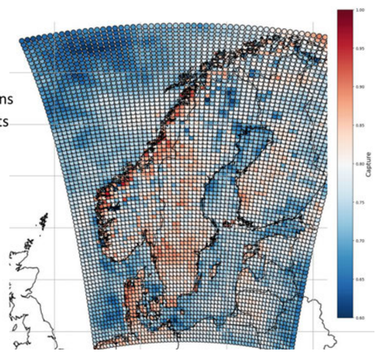
26 Leaky ReLU

- Standard ReLU activation disabled the layers in the U-Net due to dead ReLU nodes.
- Discovered through visualizing the tensors as they pass through the network



27 Verification

- 10 and 90% quantiles are much more reliable
- Still, reliability has regional patterns suggesting room for improvements



Document History

Version	Author(s)	Date	Changes
	Name (Organisation)	dd/mm/yyyy	
1.0	Jan Mirus (4cast)	23/11/2023	Initial content
1.1	Ana Prieto Nemesio (ECMWF)	26/11/2023	Main lot of content added
1.2	Peter Dueben	27/11/2023	Additional content, remarks, corrections
1.3	Jan Mirus	28/11/2023	Additional content & images added

Internal Review History

Internal Reviewers	Date	Comments
Name (Organisation)	dd/mm/yyyy	
Thomas Nipen (Norwegian Meteorological Institute)	04/12/2023	Accepted with minor revisions
Mats Brorsson (University of Luxembourg)	04/12/2023	Accepted with minor revisions

Estimated Effort Contribution per Partner

Partner	Effort
ECMWF	0.2PM
Total	**0**