Introduction to Anemoi

Jesper Dramsch

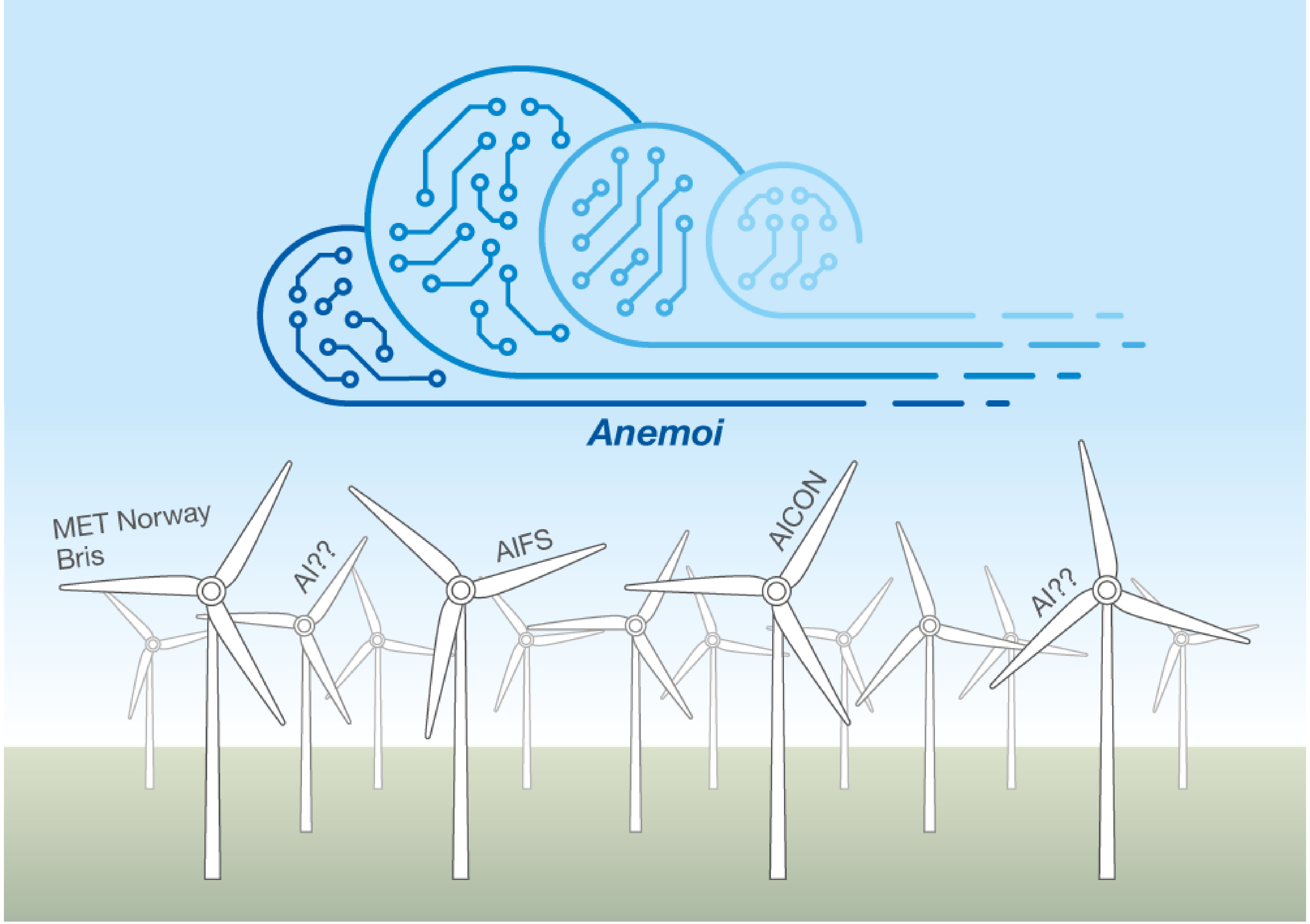
Scientist for Machine Learning

Jesper.Dramsch@ecmwf.int

Goals

- Develop an open-source framework to facilitate:
  - the development of data-driven weather forecasts…
  - … and their running in operations

- In collaboration with the Centre's Member States and others
  - Support both global and limited area models


Anemoi
MET Norway Bris
AI??
AIFS
AICON
AI??

Goals (cont.)

- Rely on existing tools
  - PyTorch, Lighting, Hydra, Zarr, Xarray, earthkit, and many more
- Focus on best use of resources (File systems, GPUs, …)
  - Do not starve the GPUs during training, due to slow I/Os
- Makes R2O as simple as possible
  - Meteorological evaluations
  - Delivery of code and weights to production

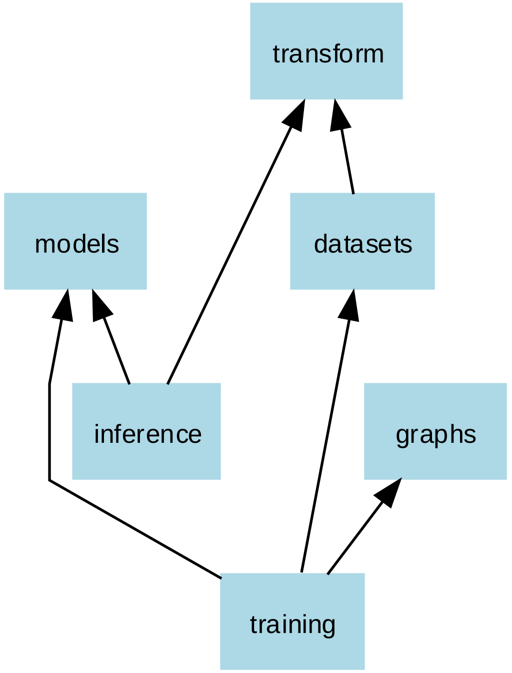
User Base

Modifies Configs for Experimentation and Improvement of Anemoi Model

Modifies Codebase to implement new Features and Augment Anemoi Libraries

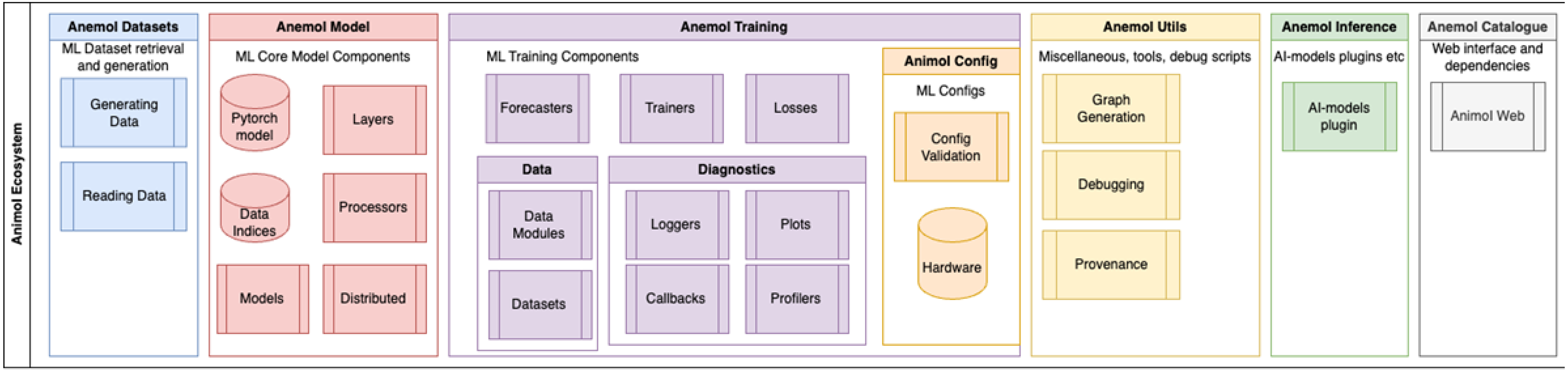
Runs the Anemoi Model in a common interface on reliable infrastructure

Main components - Design decisions



- OO design, heavy use of factories to instantiate object from config files
- Each component provides one or more command line tools
  - `anemoi-datasets create data-config.yaml out.zarr`
  - `anemoi-training train train-config.yaml`
  - `anemoi-inference run run-config.yaml`
- Minimise software dependencies to facilitate R2O
  - Inference and training are independent
- Each component collects metadata that can be used by the others

A tour through the components of Anemoi



Animol Ecosystem

Anemol Datasets

ML Dataset retrieval and generation

Generating Data

Reading Data

Anemol Model

ML Core Model Components

Pytorch model

Layers

Data Indices

Processors

Models

Distributed

Anemol Training

ML Training Components

Forecasters

Trainers

Losses

Data

Data Modules

Datasets

Diagnostics

Loggers

Plots

Callbacks

Profilers

Animol Config

ML Configs

Config Validation

Hardware

Anemol Utils

Miscellaneous, tools, debug scripts

Graph Generation

Debugging

Provenance

Anemol Inference

AI-models plugins etc

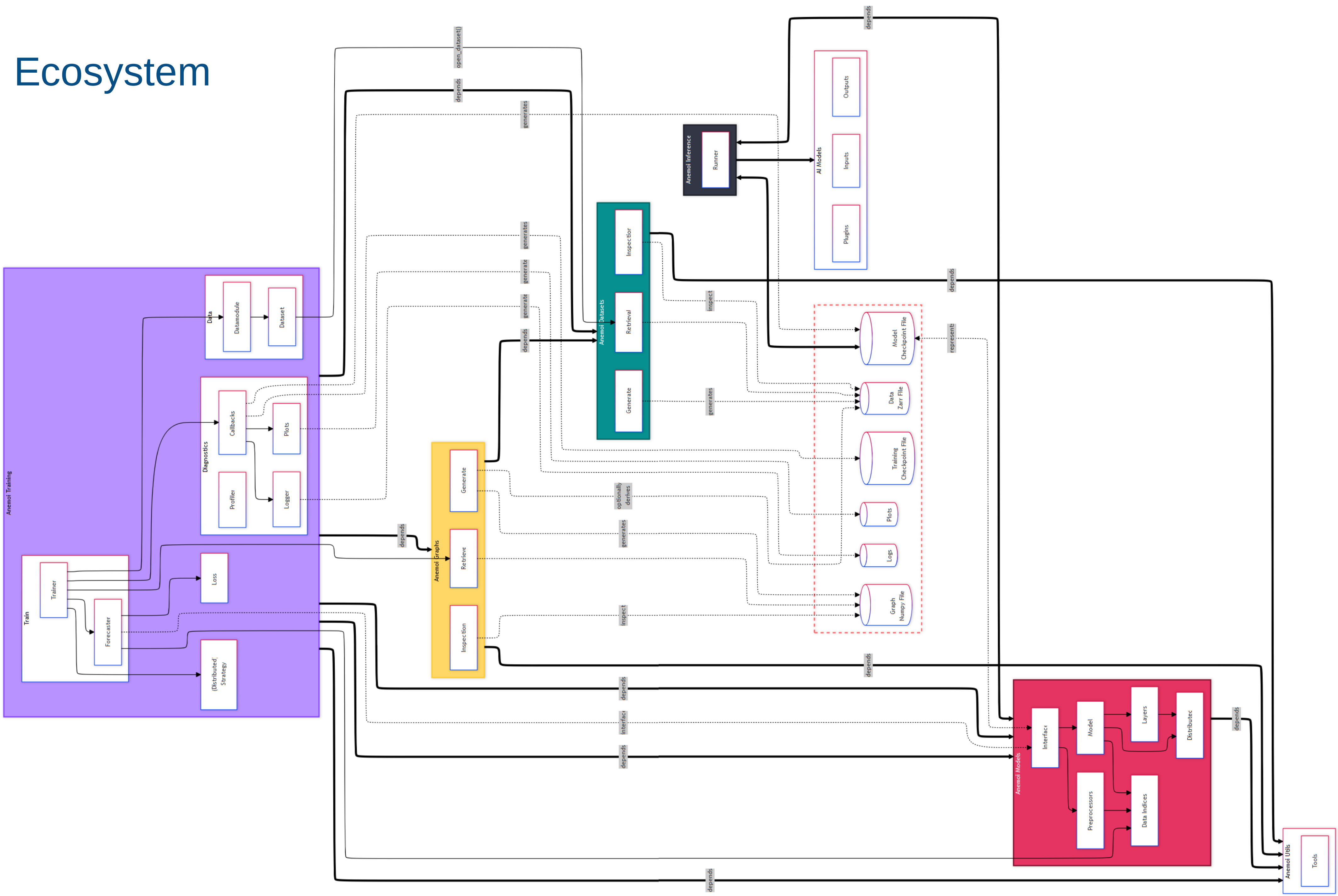
AI-models plugin

Anemol Catalogue

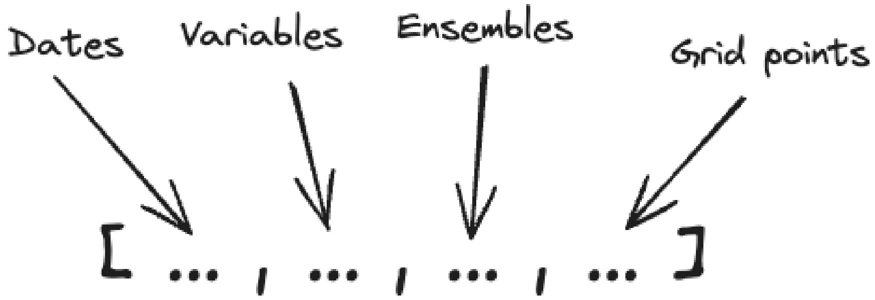
Web interface and dependencies

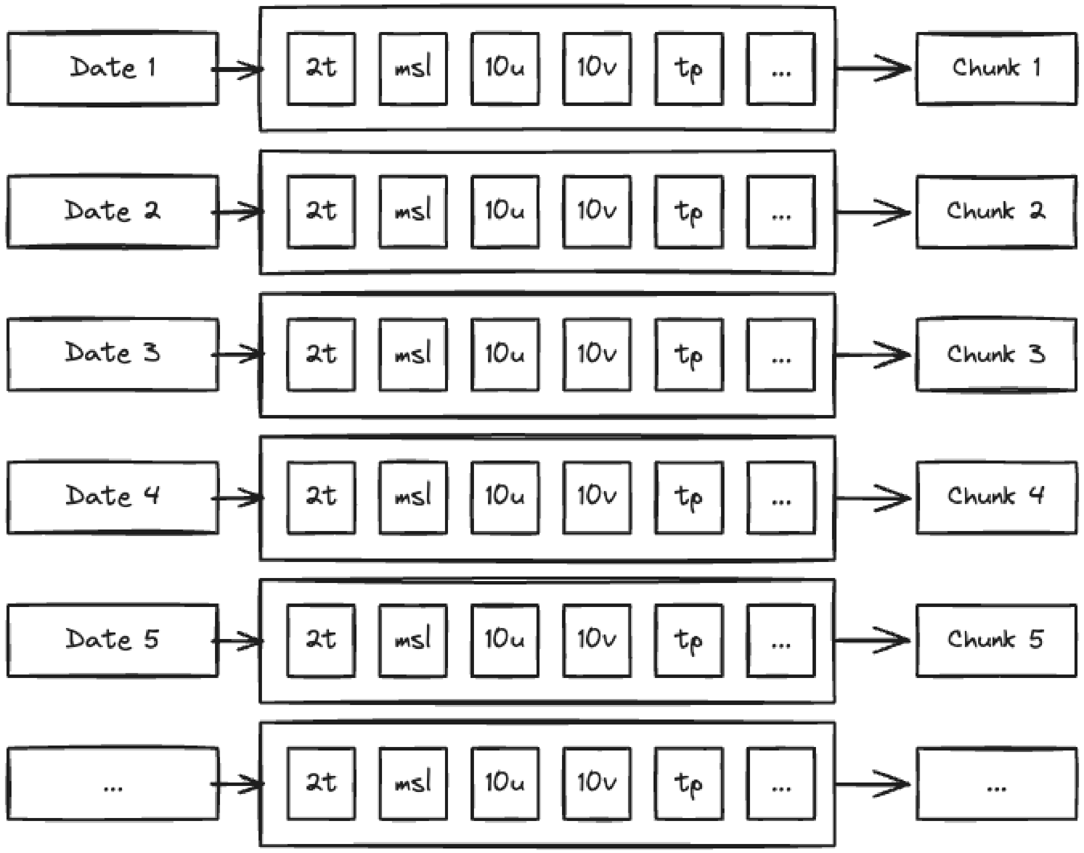
Animol Web

Ecosystem


Anemoi Training
Data
Datamodule
Dataset
Diagnostics
Callbacks
Plots
Profiler
Logger
Train
Trainer
Forecaster
Loss
(Distributed) Strategy
Anemoi Graphs
Generate
Retrieve
Inspection
Anemoi Datasets
Inspector
Retrieval
Generate
Anemoi Inference
Runner
AI Models
Outputs
Inputs
Plugins
Model Checkpoint File
Data Zarr File
Training Checkpoint File
Plots
Logs
Graph Numpy File
Anemoi Models
Interface
Model
Layers
Distributed
Preprocessors
Data Indices
Anemoi Utils
Tools
open_dataset()
depends
generates
inspect
represents
optionally derives
interface

anemoi-datasets





anemoi-datasets

Goals

- Ensuring data loading is hidden beneath training costs.
- Porting **metadata** to provide **traceability** and inference use of trained models.
- Opening Zarrs within training code.
  - Handling merges of datasets on the fly.

Features

- Creation of Zarr(s) from
  - MARS/Grib/netCDF
  - Chunk for speed, not for interpretability
- Calculation of statistics for normalisation.
- Handle multiple data sources, can be combined in a single array.

How it works:
Built on ecflow and python tools
for handling data sources.

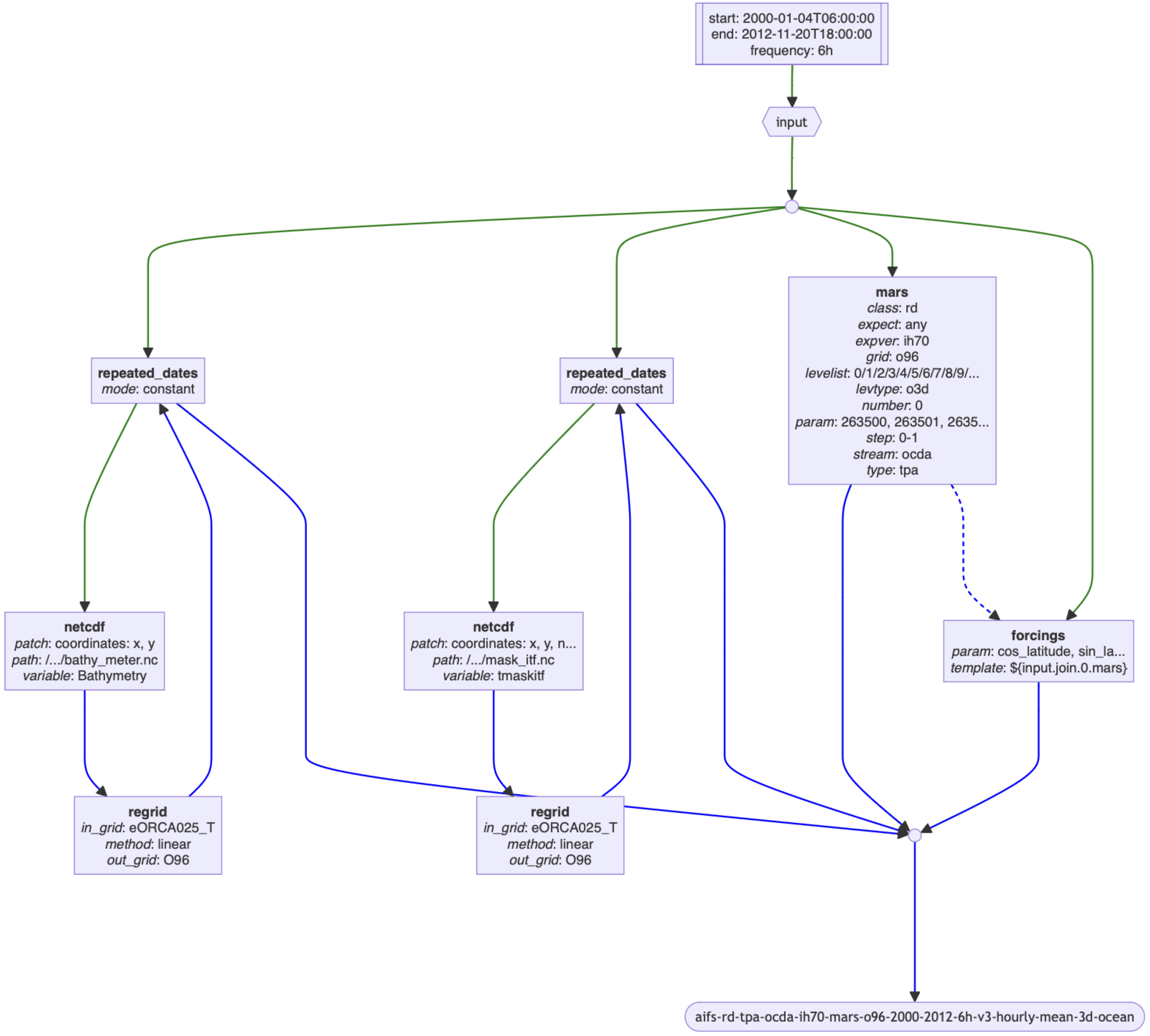
Open source, a read the docs in progress https://anemoi-datasets.readthedocs.io/

anemoi-datasets

- Tools to build Zarr files from a list of sources and filters
  - Supports GRIB, NetCDF, Zarr input format
  - Filters allows data transformations
    - e.g. wind speed/direction to u/v

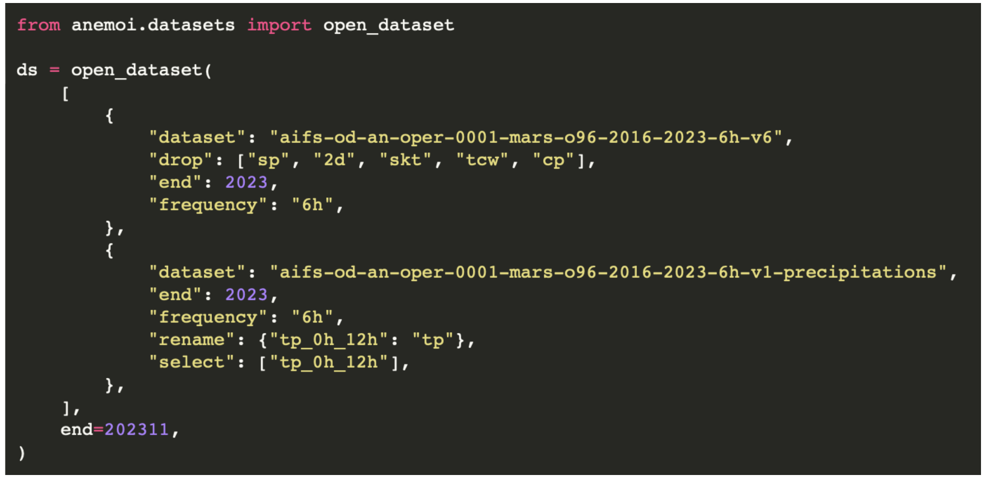
- Easily extendable with new sources and new filters
  - See anemoi-transform

- Large datasets:
  - Building can be done in parallel
  - Building can be done incrementally

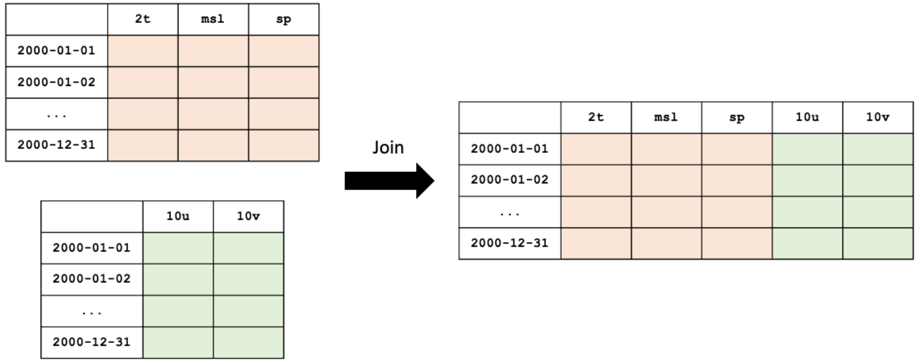


anemoi-datasets (usage)

- When used, datasets can be lazily combined into “virtual” datasets
  - Allowing researchers to find the best combinations of variables for their training needs



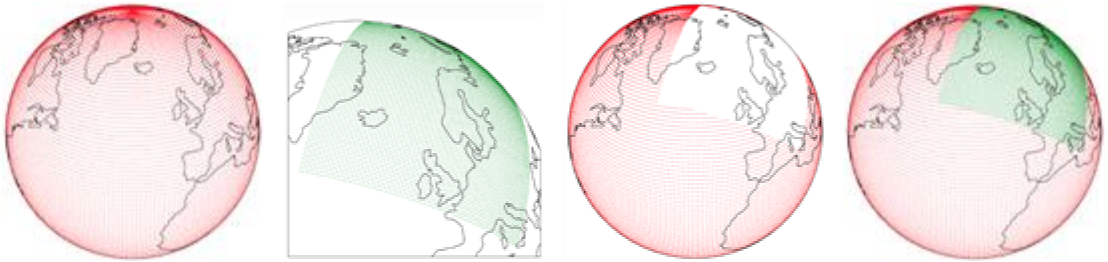
```
from anemoi.datasets import open_dataset

ds = open_dataset(
    [
        {
            "dataset": "aifs-od-an-oper-0001-mars-o96-2016-2023-6h-v6",
            "drop": ["sp", "2d", "skt", "tcw", "cp"],
            "end": 2023,
            "frequency": "6h",
        },
        {
            "dataset": "aifs-od-an-oper-0001-mars-o96-2016-2023-6h-v1-precipitations",
            "end": 2023,
            "frequency": "6h",
            "rename": {"tp_0h_12h": "tp"},
            "select": ["tp_0h_12h"],
        },
    ],
    end=202311,
)
```

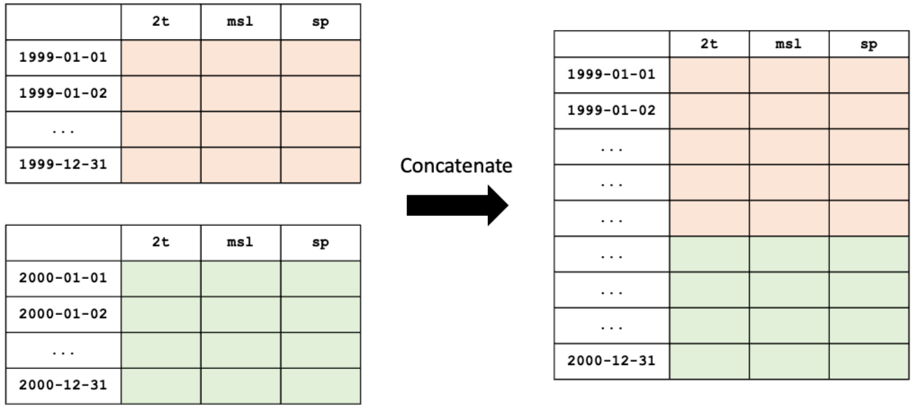


	2t	msl	sp
2000-01-01	🟧	🟧	🟧
2000-01-02	🟧	🟧	🟧
...	🟧	🟧	🟧
2000-12-31	🟧	🟧	🟧

	10u	10v
2000-01-01	🟩	🟩
2000-01-02	🟩	🟩
...	🟩	🟩
2000-12-31	🟩	🟩

	2t	msl	sp	10u	10v
2000-01-01	🟧	🟧	🟧	🟩	🟩
2000-01-02	🟧	🟧	🟧	🟩	🟩
...	🟧	🟧	🟧	🟩	🟩
2000-12-31	🟧	🟧	🟧	🟩	🟩





	2t	msl	sp
1999-01-01	🟧	🟧	🟧
1999-01-02	🟧	🟧	🟧
...	🟧	🟧	🟧
1999-12-31	🟧	🟧	🟧

	2t	msl	sp
2000-01-01	🟩	🟩	🟩
2000-01-02	🟩	🟩	🟩
...	🟩	🟩	🟩
2000-12-31	🟩	🟩	🟩

	2t	msl	sp
1999-01-01	🟧	🟧	🟧
1999-01-02	🟧	🟧	🟧
...	🟧	🟧	🟧
...	🟧	🟧	🟧
...	🟧	🟧	🟧
...	🟩	🟩	🟩
...	🟩	🟩	🟩
...	🟩	🟩	🟩
2000-12-31	🟩	🟩	🟩

anemoi-training

- Code to train models, using torch-lightning and Hydra
  - Multi-node/multi-GPU training support
  - Deterministic training with probabilistic training coming soon…
  - Callbacks for profiling evaluating, plotting and logging intermediate results
  - Implement various losses, more can be easily added
  - Interfaces with trackers such as mlflow

- Highly configurable

- Interfaces with:
  - anemoi-dataset via data loaders
  - anemoi-models via Hydra configuration
  - anemoi-inference via metadata-rich checkpoints

anemoi-training (cont.)

```
Model / GNN.yml
activation: GELU
num_channels: 512

model:
  _target_:
anemoi.models.models.encoder_processor_decoder
.AnemoiModelEncProcDec

processor:
  _target_:
anemoi.models.layers.processor.GNNProcessor
  _convert_: all
  activation: ${model.activation}
  trainable_size:
${model.trainable_parameters.hidden2hidden}
  sub_graph_edge_attributes:
${model.attributes.edges}
  num_layers: 16
  num_chunks: 2
  mlp_extra_layers: 0

encoder:
  _target_:
anemoi.models.layers.mapper.GNNForwardMapper
```

```
anemoi-training train model=gnn
```

```
Model / GraphTransformer.yml
activation: GELU
num_channels: 1024

model:
  _target_:
anemoi.models.models.encoder_processor_decoder.Ane
moiModelEncProcDec

processor:
  _target_:
anemoi.models.layers.processor.GraphTransformerPro
cessor
  _convert_: all
  activation: ${model.activation}
  trainable_size:
${model.trainable_parameters.hidden2hidden}
  sub_graph_edge_attributes:
${model.attributes.edges}
  num_layers: 16
  num_chunks: 2
  mlp_hidden_ratio: 4 # GraphTransformer
  num_heads: 16 # GraphTransformer

encoder:
```

```
anemoi-training train model=graphtransformer
```

```
Model / Transformer.yml
activation: GELU
num_channels: 1024

model:
  _target_:
anemoi.models.models.encoder_processor_decoder.AnemoiModelEn
cProcDec

processor:
  _target_:
anemoi.models.layers.processor.TransformerProcessor
  _convert_: all
  activation: ${model.activation}
  num_layers: 16
  num_chunks: 2
  mlp_hidden_ratio: 4 # Transformer only
  num_heads: 16 # Transformer only
  window_size: 512
  dropout_p: 0.0

encoder:
 _target_: anemoi.models.layers.mapper.GraphTransformerFor
```

```
anemoi-training train model=transformer
```

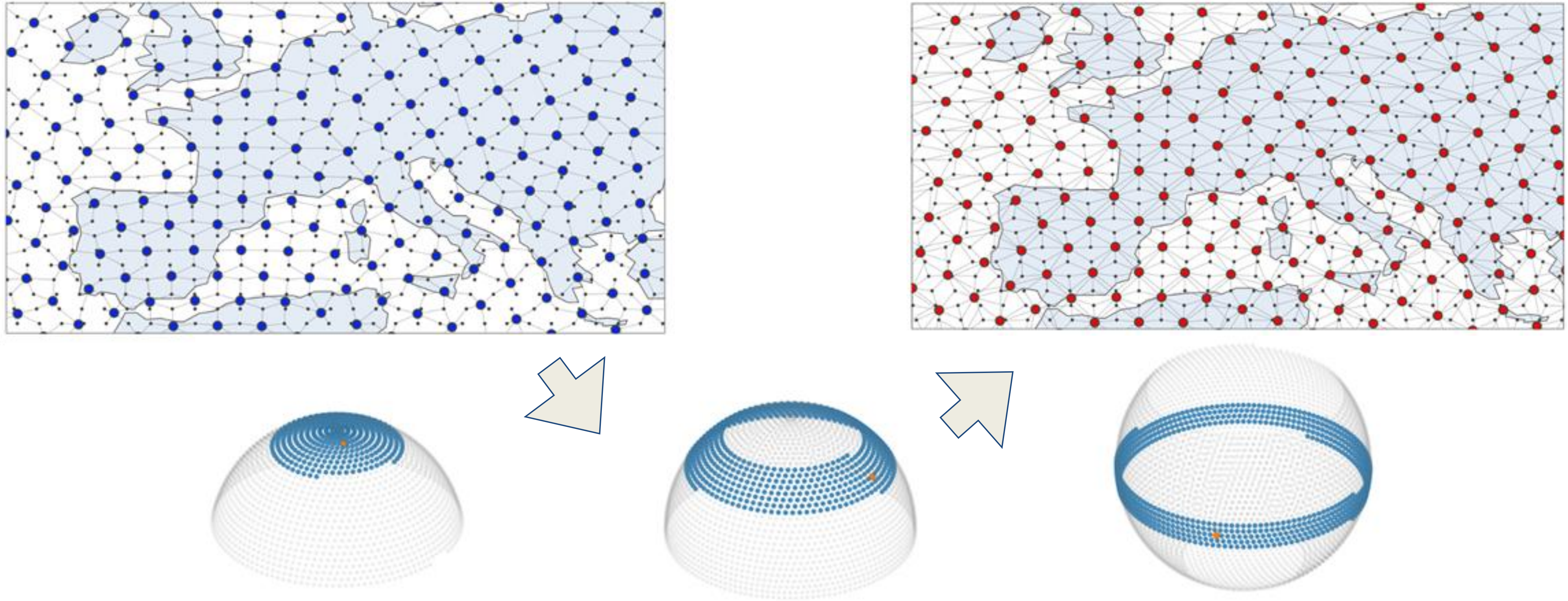
- Make it easy to switch components
- Allow for reproduceable training
- Easy to extend with new models and components

anemoi-training

Example: current AIFS configuration

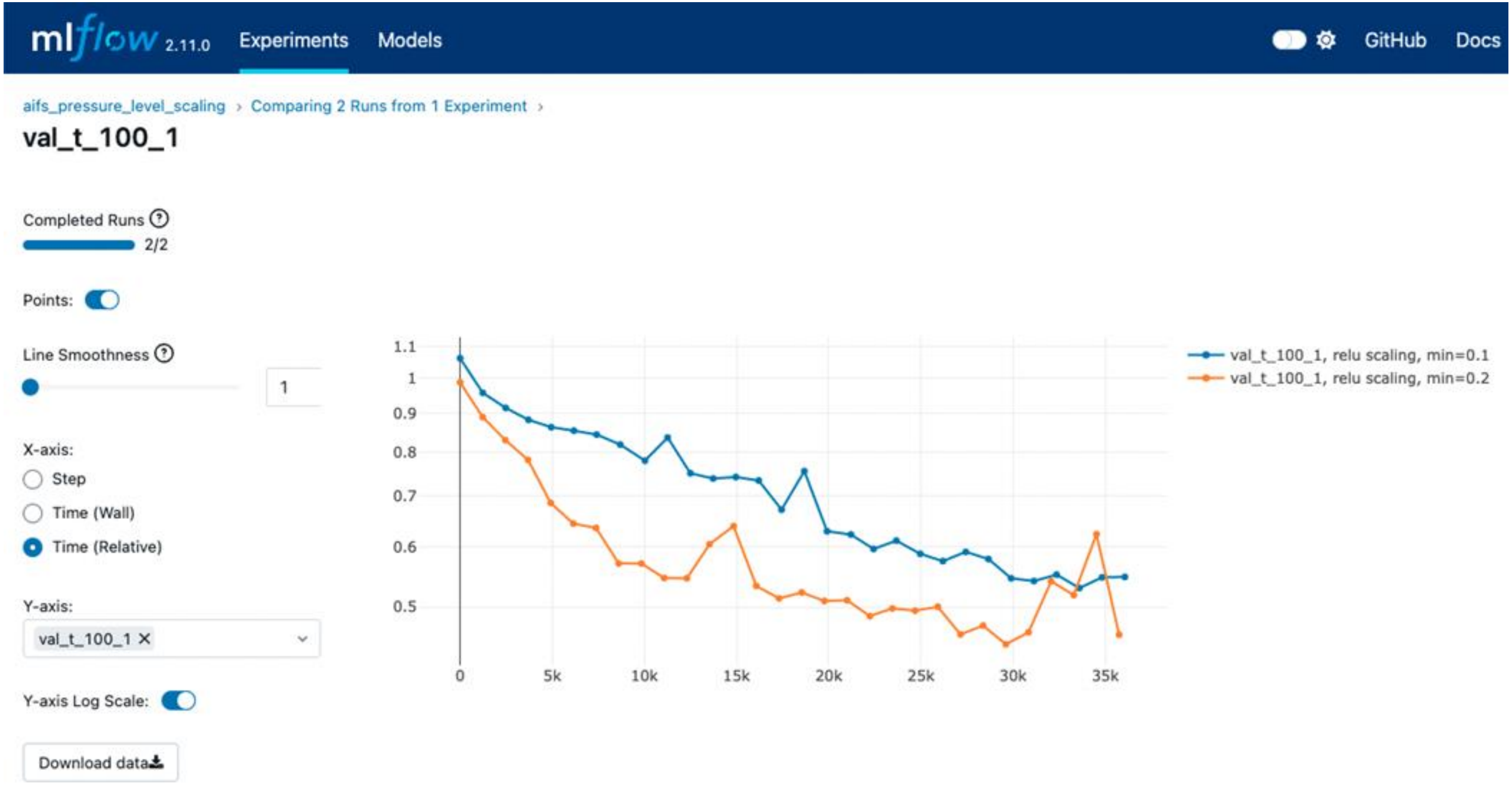
Graph Encoder

Graph Decoder



Transformer processor

Anemoi.training – MLFlow servers for Experiment Tracking



aifs_pressure_level_scaling › Comparing 2 Runs from 1 Experiment ›

val_t_100_1

Completed Runs 2/2

Points: ●

Line Smoothness 1

X-axis:

○ Step

○ Time (Wall)

● Time (Relative)

Y-axis: val_t_100_1 ✕

Y-axis Log Scale: ●

Download data

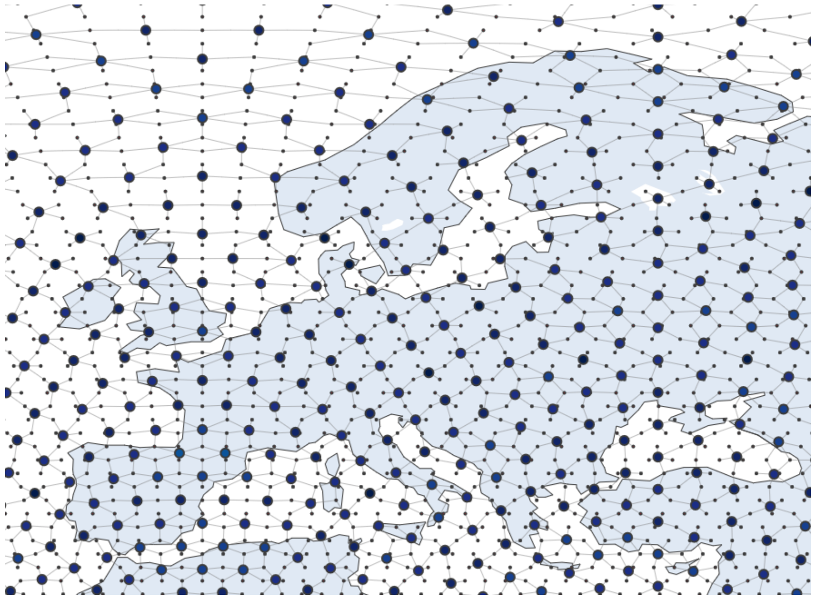
1.1
1
0.9
0.8
0.7
0.6
0.5
0
5k
10k
15k
20k
25k
30k
35k
val_t_100_1, relu scaling, min=0.1
val_t_100_1, relu scaling, min=0.2

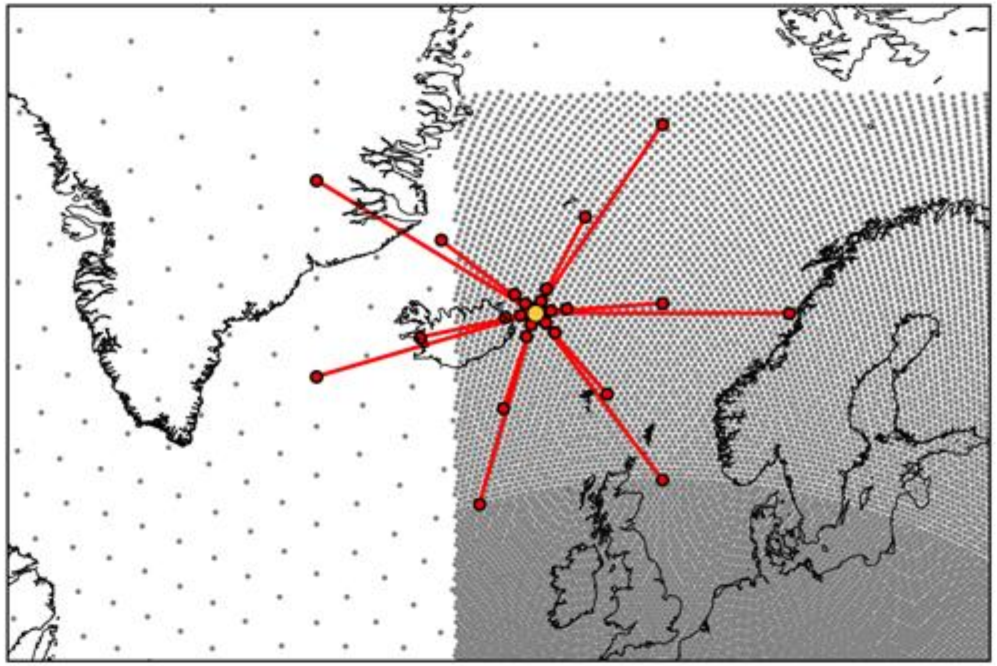
anemoi-models

- Building blocks to implement data-driven forecasting systems
  - Models
  - Blocks (e.g. encoders/decoders/processors)
  - Layers
- Design choice: graph-based (or similar) outer layers.
  - Provides high levels of flexibility on input & output data.
  - Inner layers/blocks can by either attention- or graph-based.
- Multi-node, multi-GPUs to scale to large model
  - Multiple model-sharded implementations supported, e.g. 1-hop graph, head sharding.
  - Ensures model design is not limited by GPU-memory.
- Supports GraphCast-style, Oskarson-style and AIFS models.
- Further models can be easily created

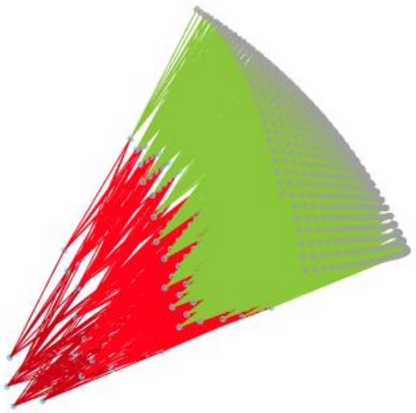
anemoi-graphs

- Highly configurable
- Integrated with Anemoi ecosystem
- Includes graph inspection tools
- Optimized for GPU
- Supports multiple applications:
  - Global weather forecasting
  - Limited area modelling
  - Stretched-grid graphs
  - Hierarchical graphs
  - Dynamic graphs in progress

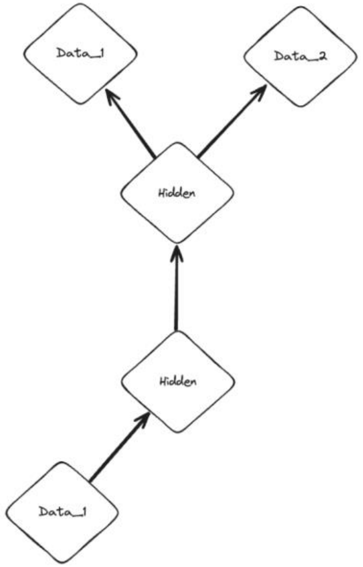




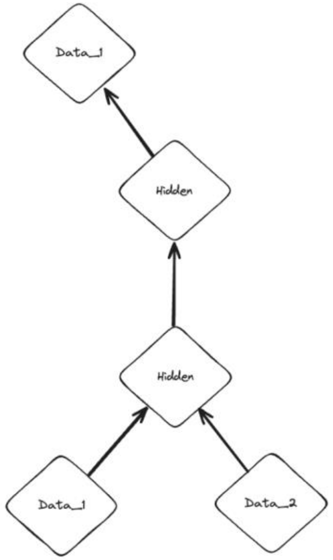
What comes next?



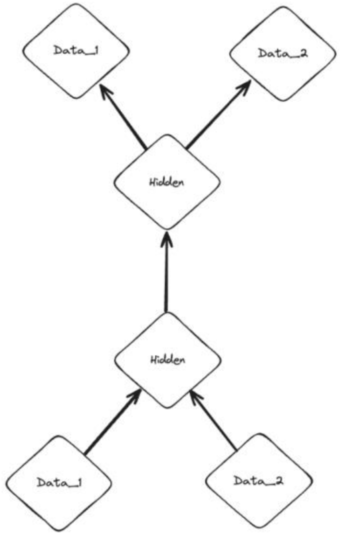
Hierarchical graph



Multiple decoder

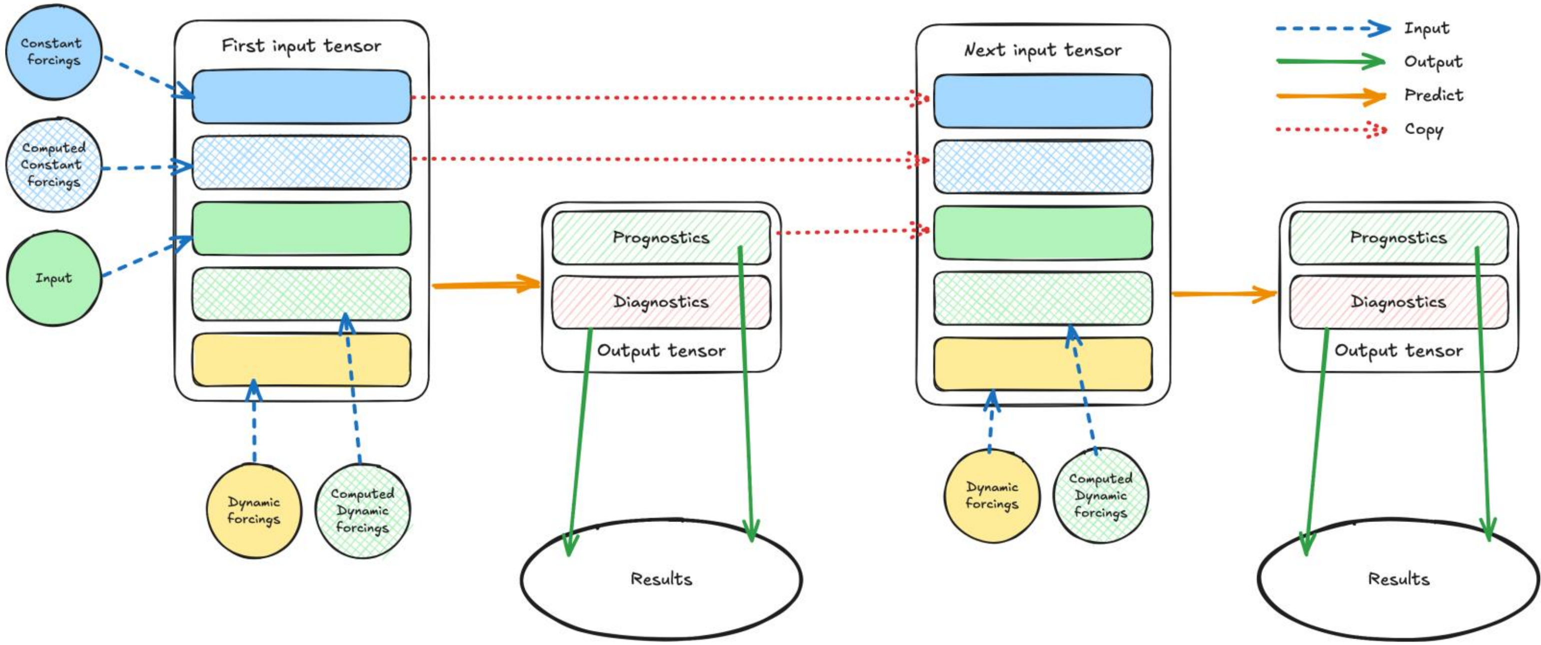


Multiple encoder



Multiple encoder & decoder

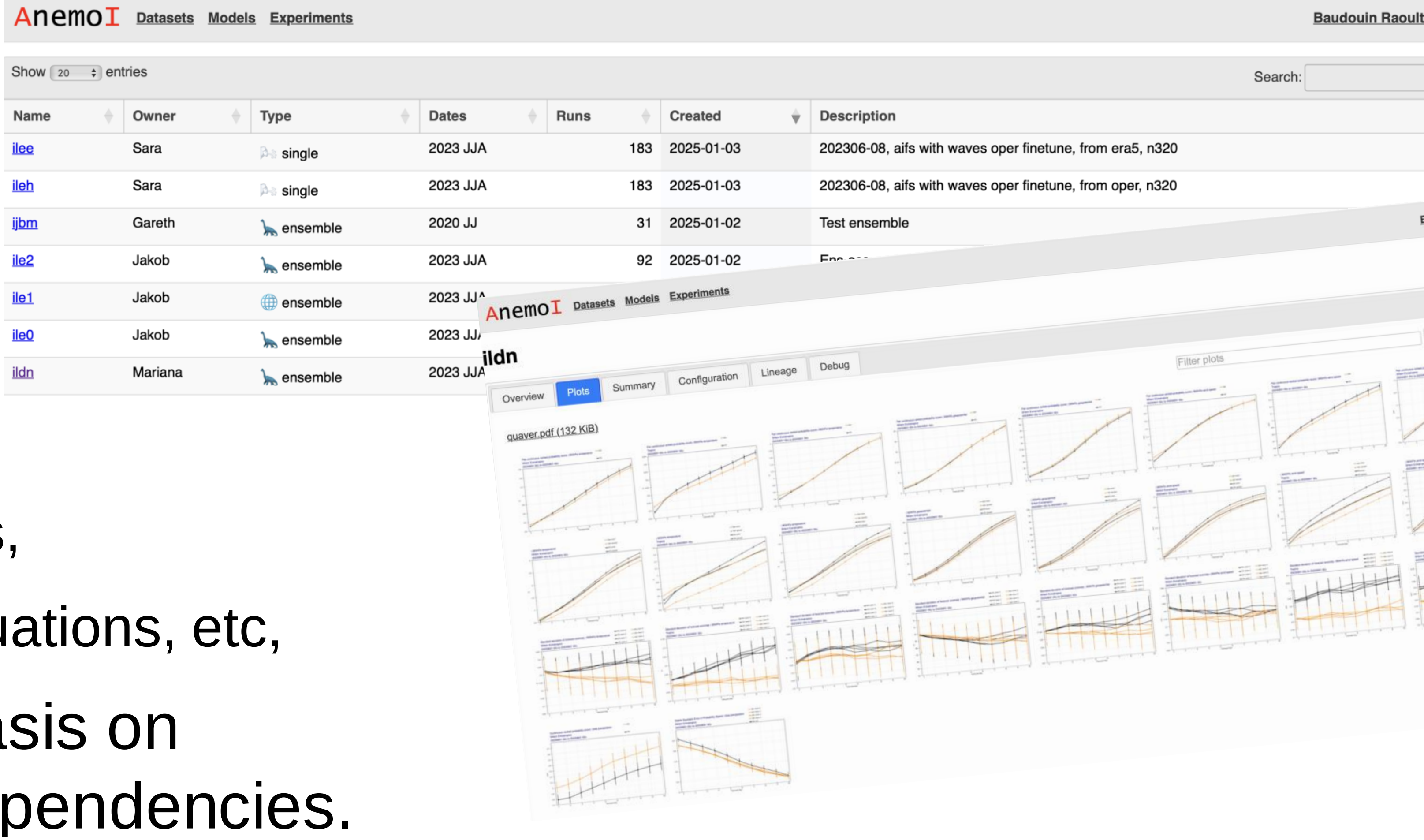
anemoi-inference



- Running trained models in inference mode
- Auto-discovery of the model parameters from the metadata embedded in the checkpoint
  - List of prognostic, diagnostic and forcing variables
- Provide full control of input and output to the user
  - E.g. using the output of physics-based NWP analyses or ensembles, as initial conditions
- In progress: allowing coupling of two of more data-driven forecasts
  - E.g. ocean/atmosphere

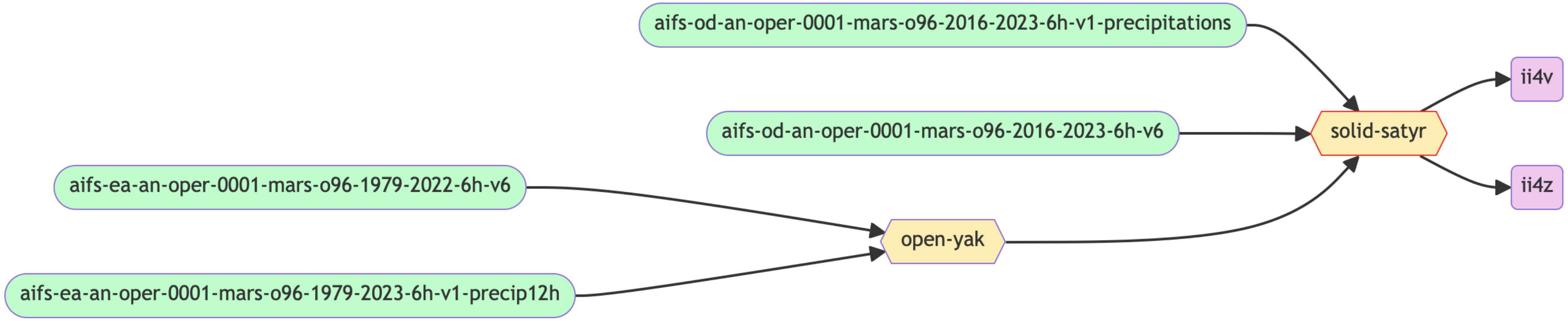
anemoi-registry

- code to manage a catalogue
  - training datasets,
  - model configurations,
  - weights, model evaluations, etc,
- With a strong emphasis on recording lineage dependencies.



Name	Owner	Type	Dates	Runs	Created	Description
ilee	Sara	single	2023 JJA	183	2025-01-03	202306-08, aifs with waves oper finetune, from era5, n320
ileh	Sara	single	2023 JJA	183	2025-01-03	202306-08, aifs with waves oper finetune, from oper, n320
ijbm	Gareth	ensemble	2020 JJ	31	2025-01-02	Test ensemble
ile2	Jakob	ensemble	2023 JJA	92	2025-01-02	Ens
ile1	Jakob	ensemble	2023 JJ			
ile0	Jakob	ensemble	2023 JJ			
ildn	Mariana	ensemble	2023 JJA			

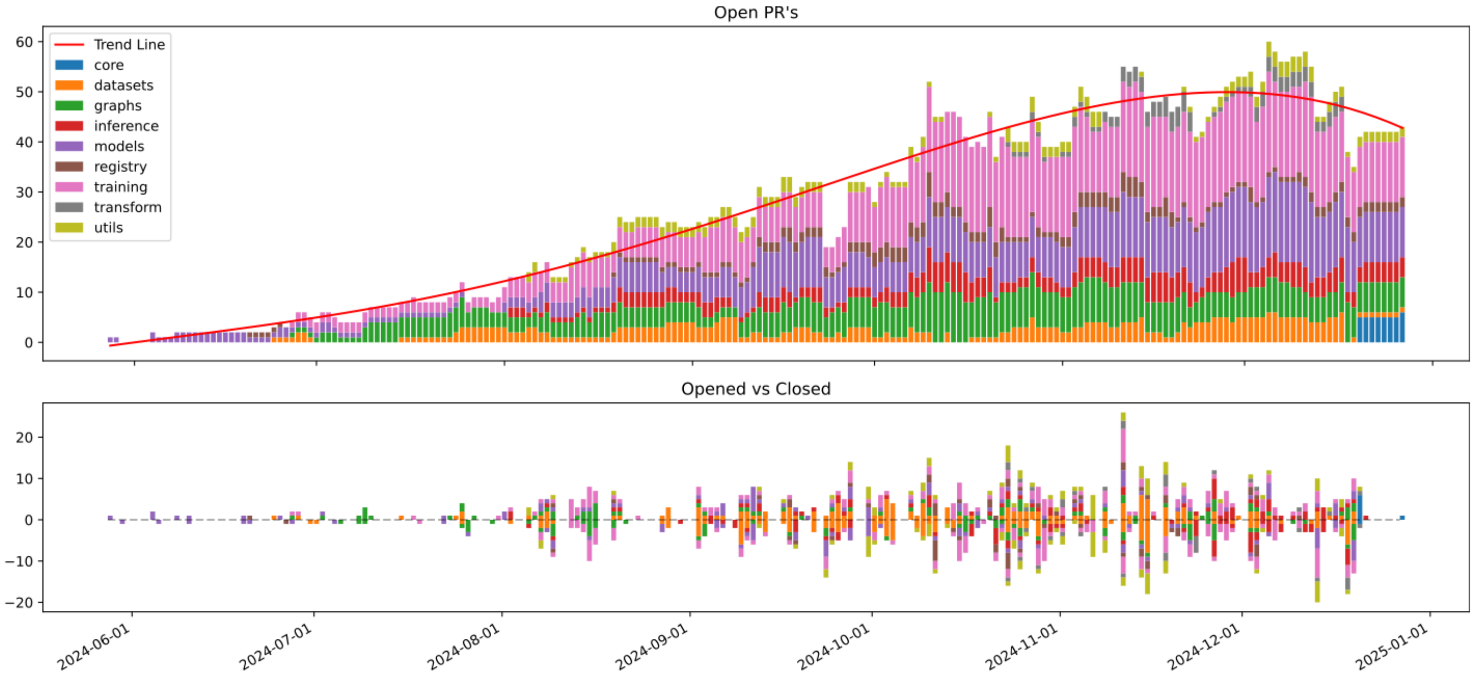
AnemoI
Datasets
Models
Experiments
ildn
Overview
Plots
Summary
Configuration
Lineage
Debug
Filter plots
quaver.pdf (132 KiB)



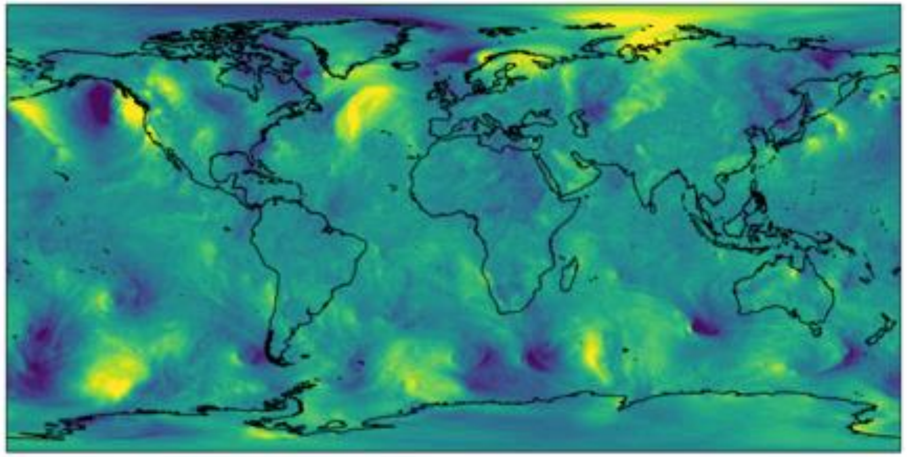
aifs-od-an-oper-0001-mars-o96-2016-2023-6h-v1-precipitations
aifs-od-an-oper-0001-mars-o96-2016-2023-6h-v6
aifs-ea-an-oper-0001-mars-o96-1979-2022-6h-v6
aifs-ea-an-oper-0001-mars-o96-1979-2023-6h-v1-precip12h
open-yak
solid-satyr
ii4v
ii4z

Collaboration

- Code reviews via pull requests
- Pre-commit checks
- GitHub actions
- CI/CD
- Anyone can contribute!



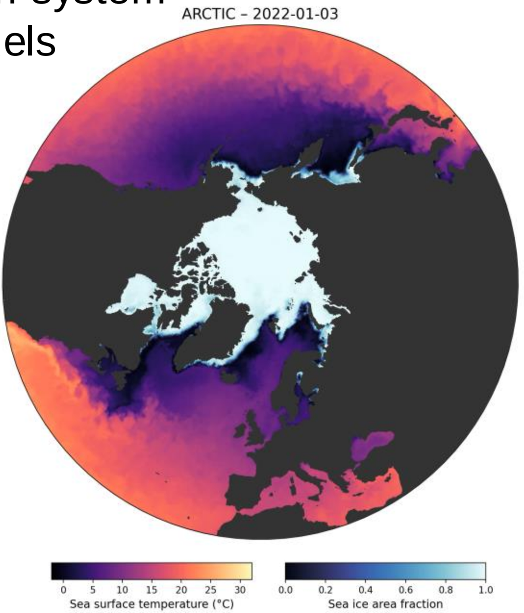
What Anemoi enables?

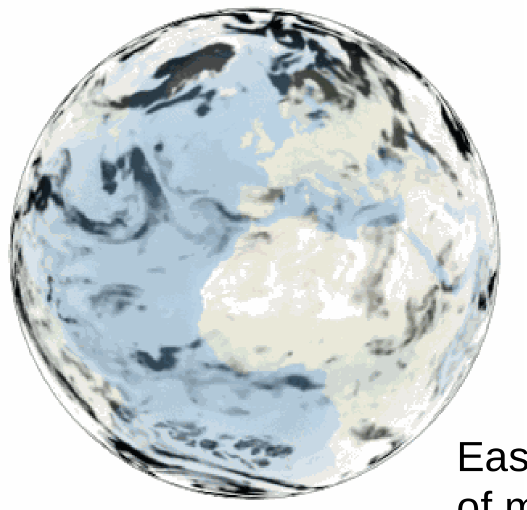


Training deterministic & ensemble models

Lang et al. 2024a & b

Building coupled earth-system models



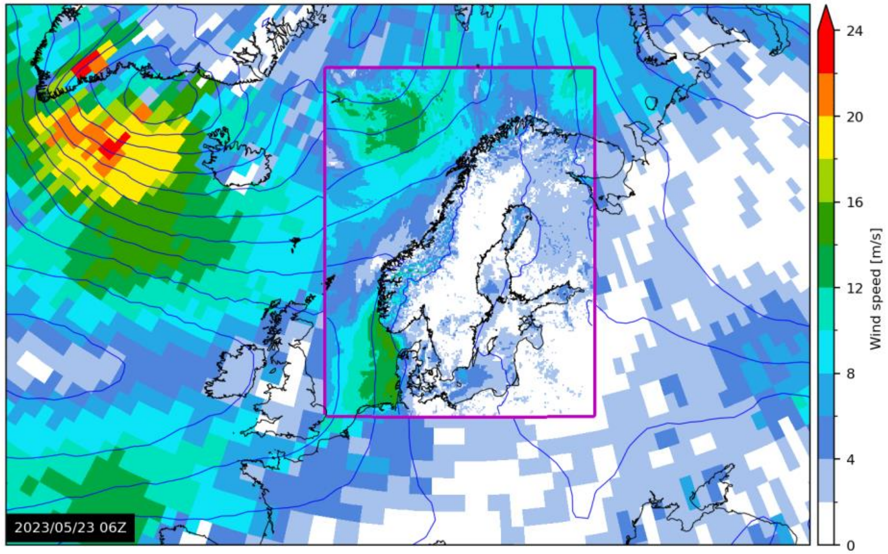


Easy incorporation of more atmospheric fields

Nipen et al. 2024

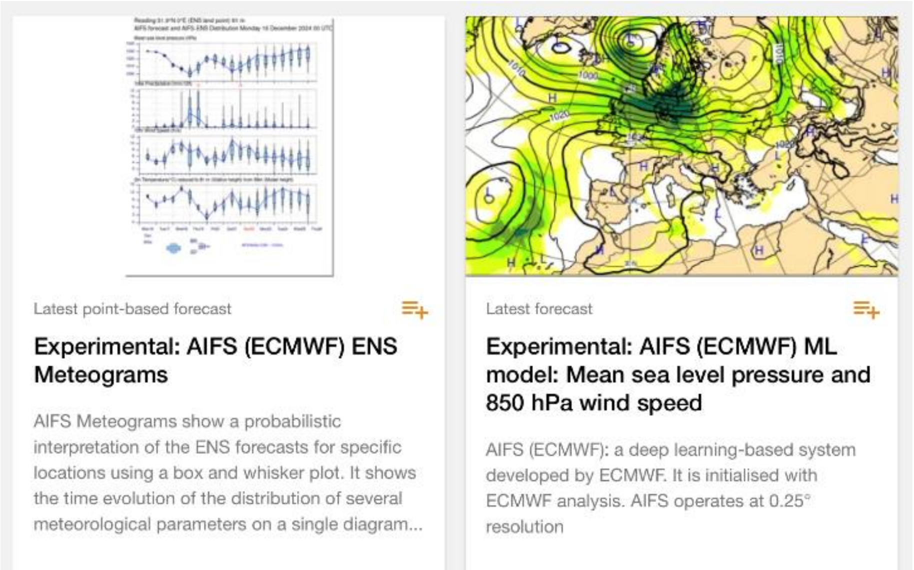
Regional high-resolution modelling:

See Bris poster (970) on Thursday afternoon



Real-time running at operation centres

Anemoi will power ECMWF’s operationalisation of AIFS in 2025.



Latest point-based forecast

Experimental: AIFS (ECMWF) ENS Meteograms

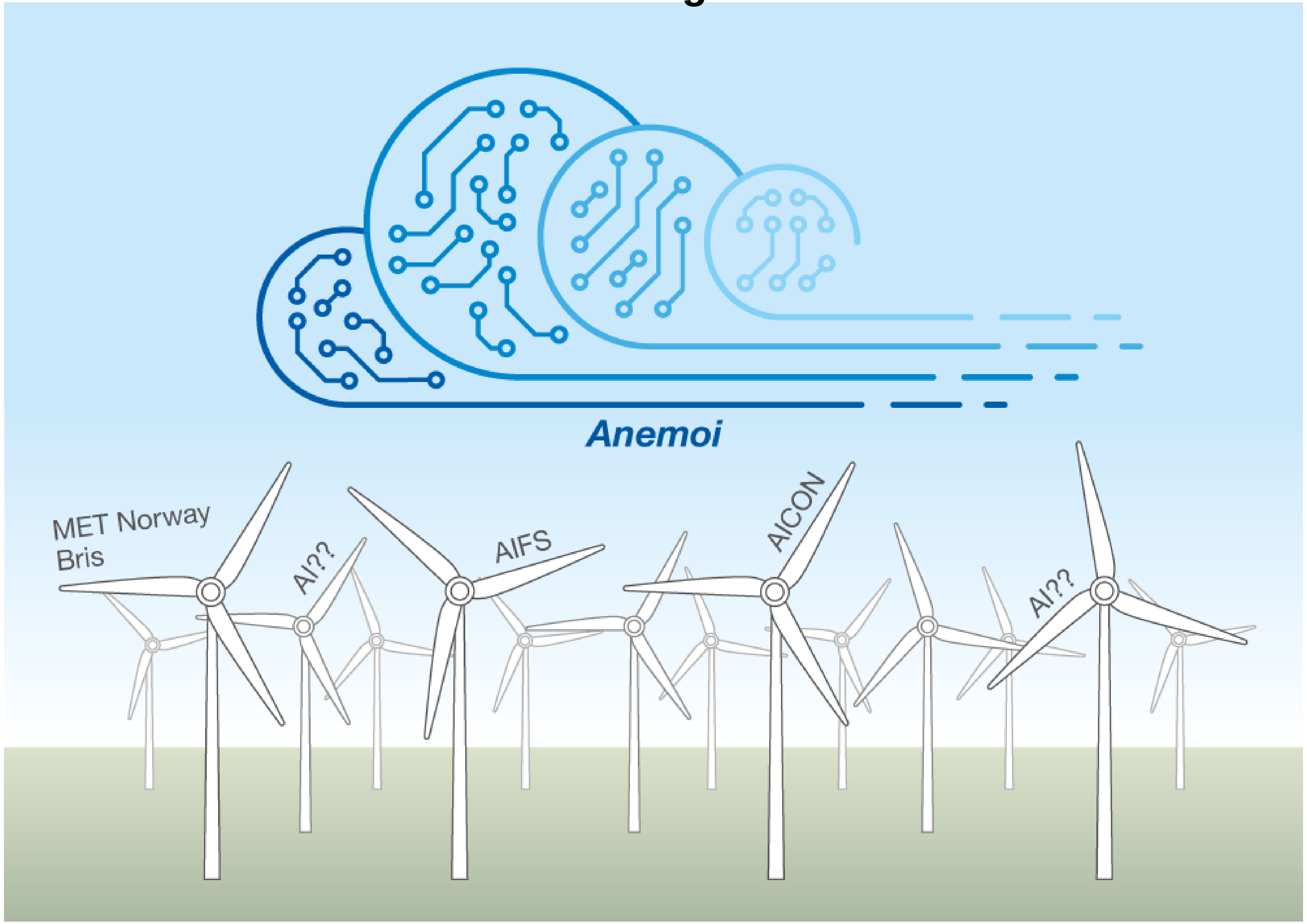
AIFS Meteograms show a probabilistic interpretation of the ENS forecasts for specific locations using a box and whisker plot. It shows the time evolution of the distribution of several meteorological parameters on a single diagram...

Latest forecast

Experimental: AIFS (ECMWF) ML model: Mean sea level pressure and 850 hPa wind speed

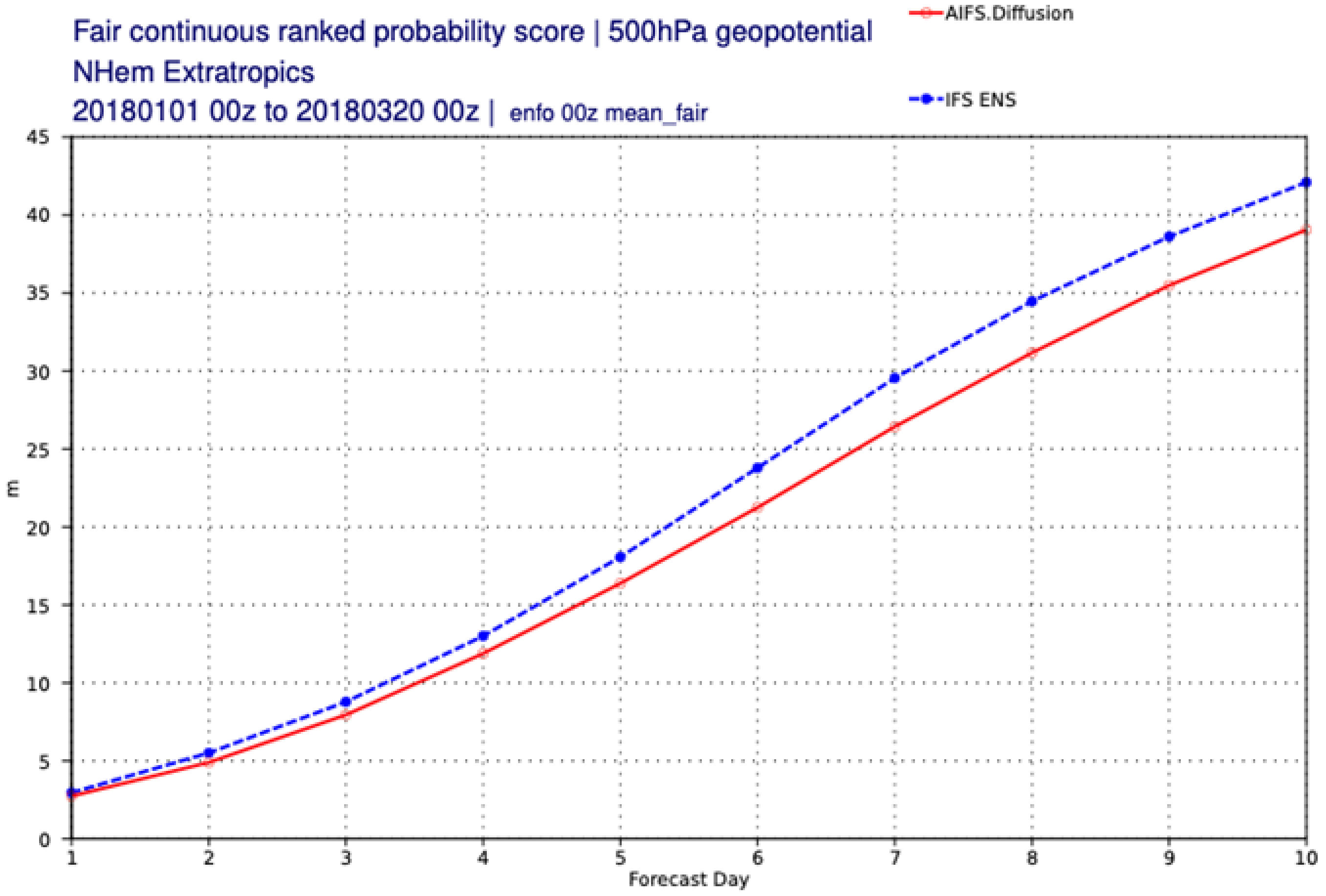
AIFS (ECMWF): a deep learning-based system developed by ECMWF. It is initialised with ECMWF analysis. AIFS operates at 0.25° resolution

Collaboration, without aggregation around a single model



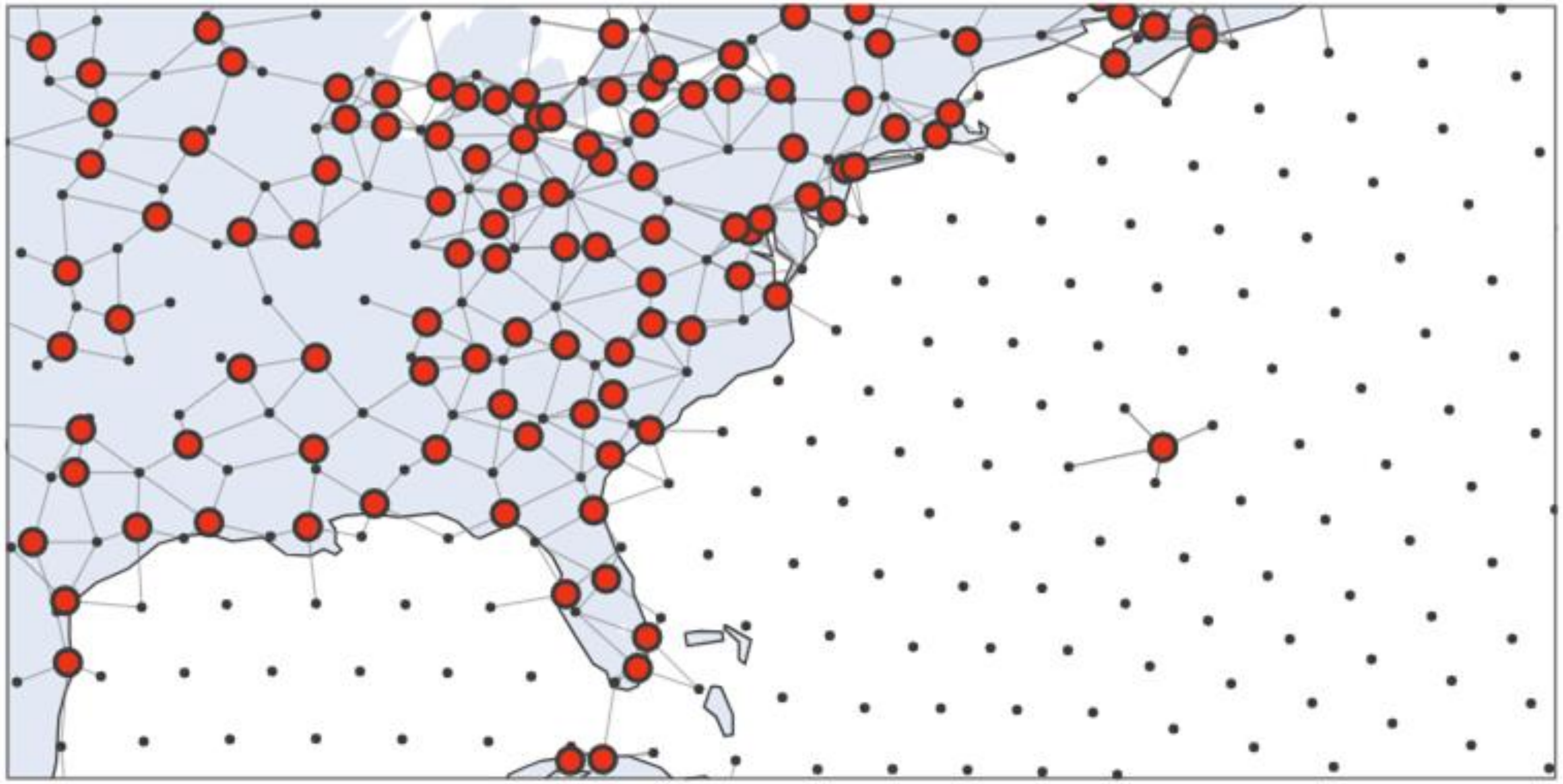
Roadmap – what’s coming next for Anemoi

- Ensembles, supporting score optimisation & diffusion training



Roadmap – what's coming next for Anemoi

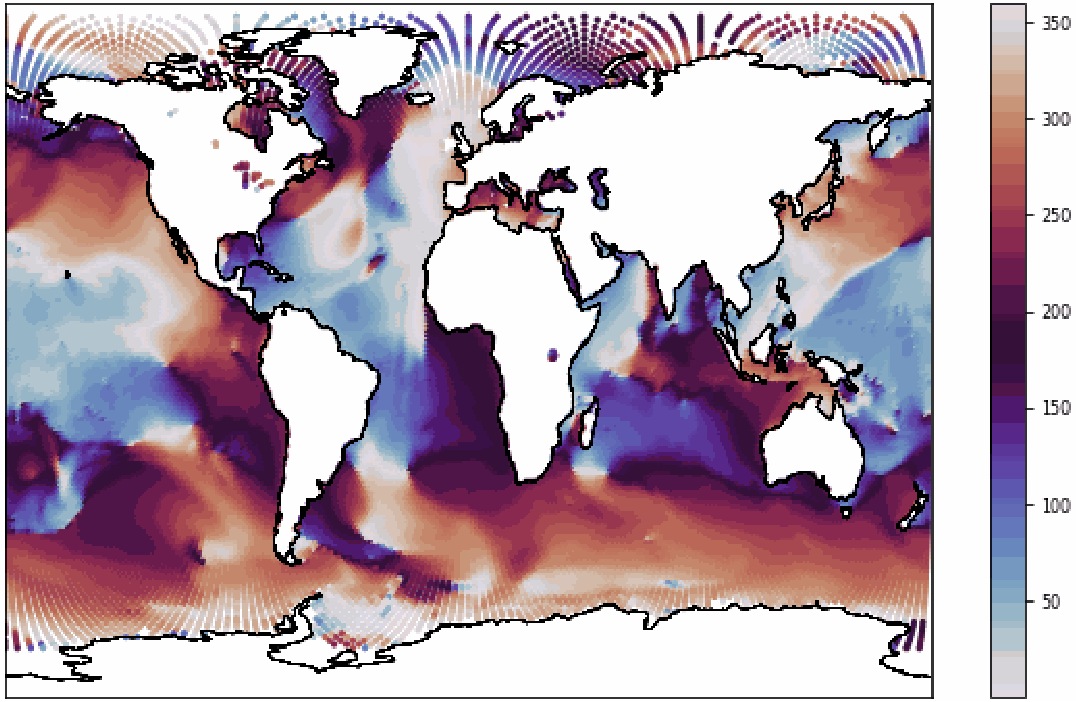
- Dynamic graphs for training towards observations

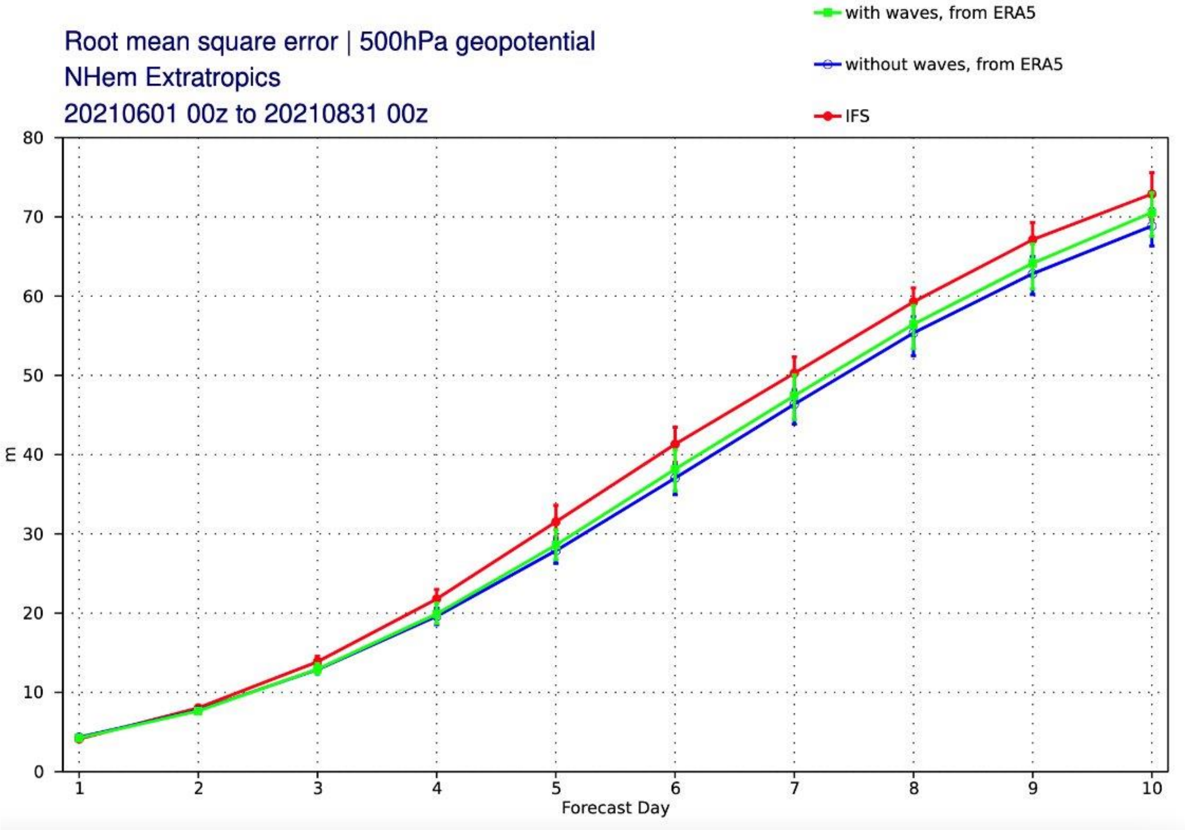


Roadmap – Earth system components

- Preliminary Result: 14-day forecast of wave fields without deteriorating the performance in the atmosphere (o96)

Mean wave direction (14 day forecast)





the lower the better

What’s next?

- Ensembles: both CRPS-optimization and diffusion (similar to GenCast)
- Better support for observations
- GPU-parallel inference
- Support for environmental data (ocean, land, waves)
- Global/LAM coupling
- Automated training and validation

Thank you