Going Global

Taking code from **research** to **operational** open ecosystem for AI **weather forecasting**

ECMWF | EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

Dr. Jesper Dramsch

Thanks to all contributors to AIFS and Anemoi

Rilwan Adewoyin, Mihai Alexe, Zied Ben Bouallègue, Matthew Chantry, Mariana Clare, Harrison Cook, Jesper Dramsch, Joffrey Dumont Le Brazidec, Rachel Furner, Vera Gahlen, Sara Hahner, Aaron Hopkinson, Gareth Jones, Simon Lang, Christian Lessig, Martin Leutbecher, Linus Magnusson, Michael Maier-Gerber, Gert Mertes, Gabriel Moldovan, Ana Prieto Nemesio, Cathal O'Brien, Florian Pinault, Ewan Pinnington, Jan Polster, Baudouin Raoult, Nina Raoult, Mario Santa Cruz, Jakob Schloer, Maria Luisa Taccari, Helen Theissen, Steffen Tietsche, Lorenzo Zampieri

Developed and used by meteorological centers across Europe.

AEMET, DWD, FMI, GeoSphere, KNMI, MET Norway, Meteo Swiss, Meteo France, RMI, & ECMWF

Goals for this talk

- Growing software projects
- Anticipating user needs
- Learn about weather forecasting
- AI Cautionary Tales
- Upskilling From Coder to Software Architect
- Planning for features you can't even know about
- Growing a team to significant size
- Have fun

Who am I?

Scientist for Machine Learning in Weather Forecasting

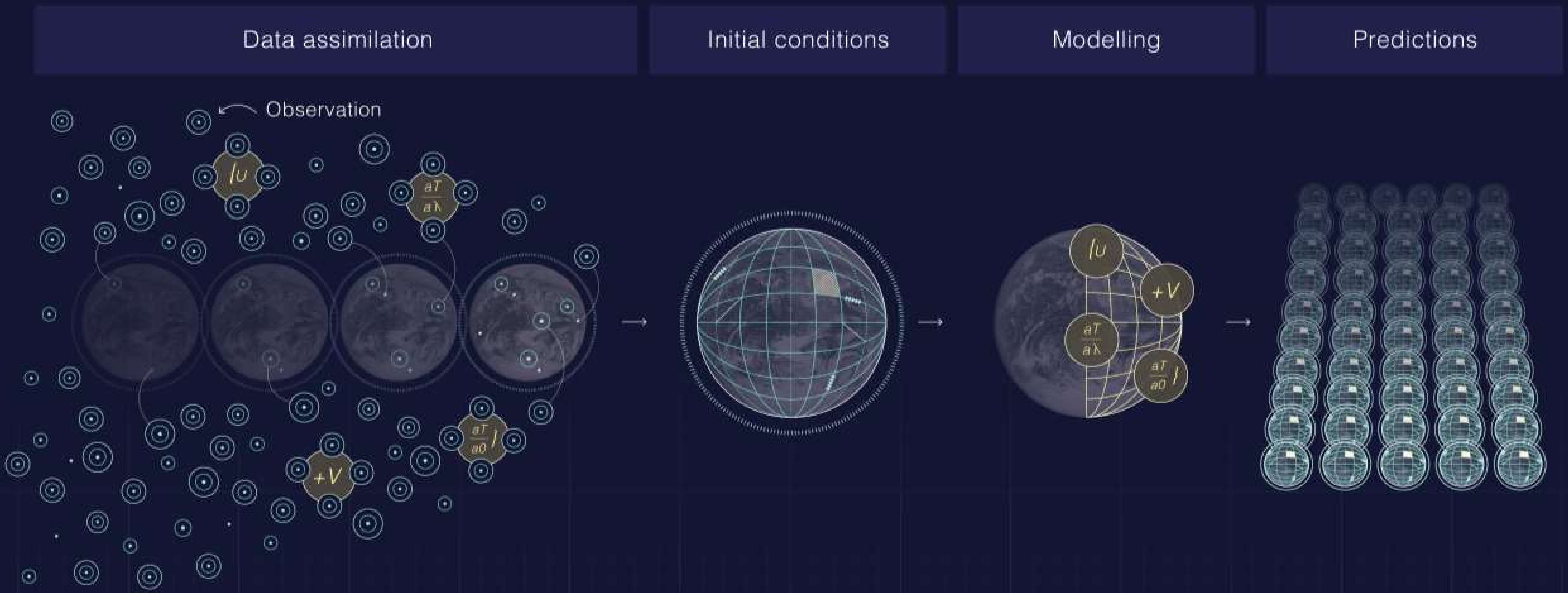
PhD in Machine Learning for Geoscience

Python, Data, AI Education

Maintain Pythondeadlin.es, ML.recipes, data-science-gui.de

Generally loud online

CC-BY Theo Wadorno

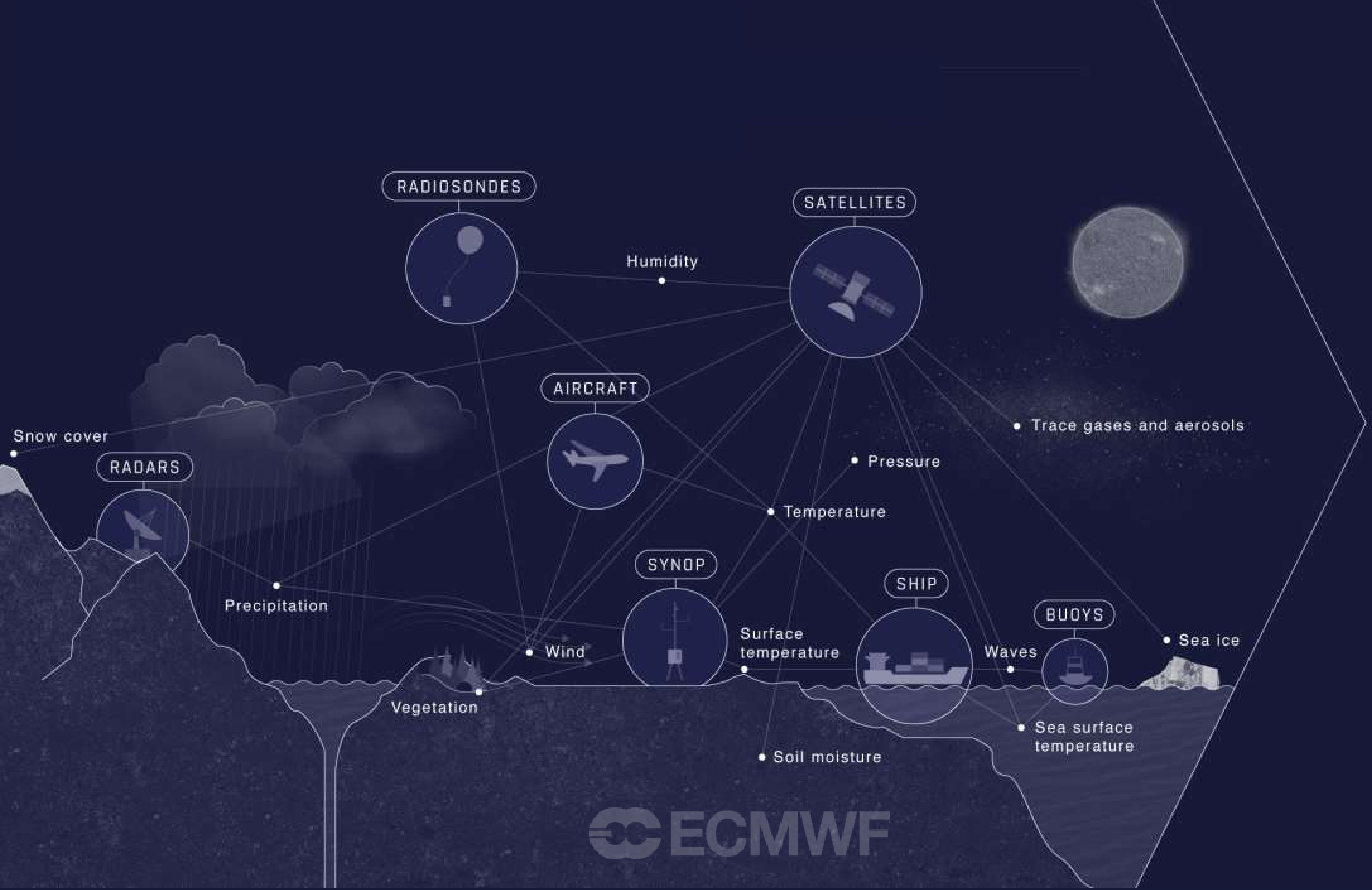


Data assimilation

Initial conditions

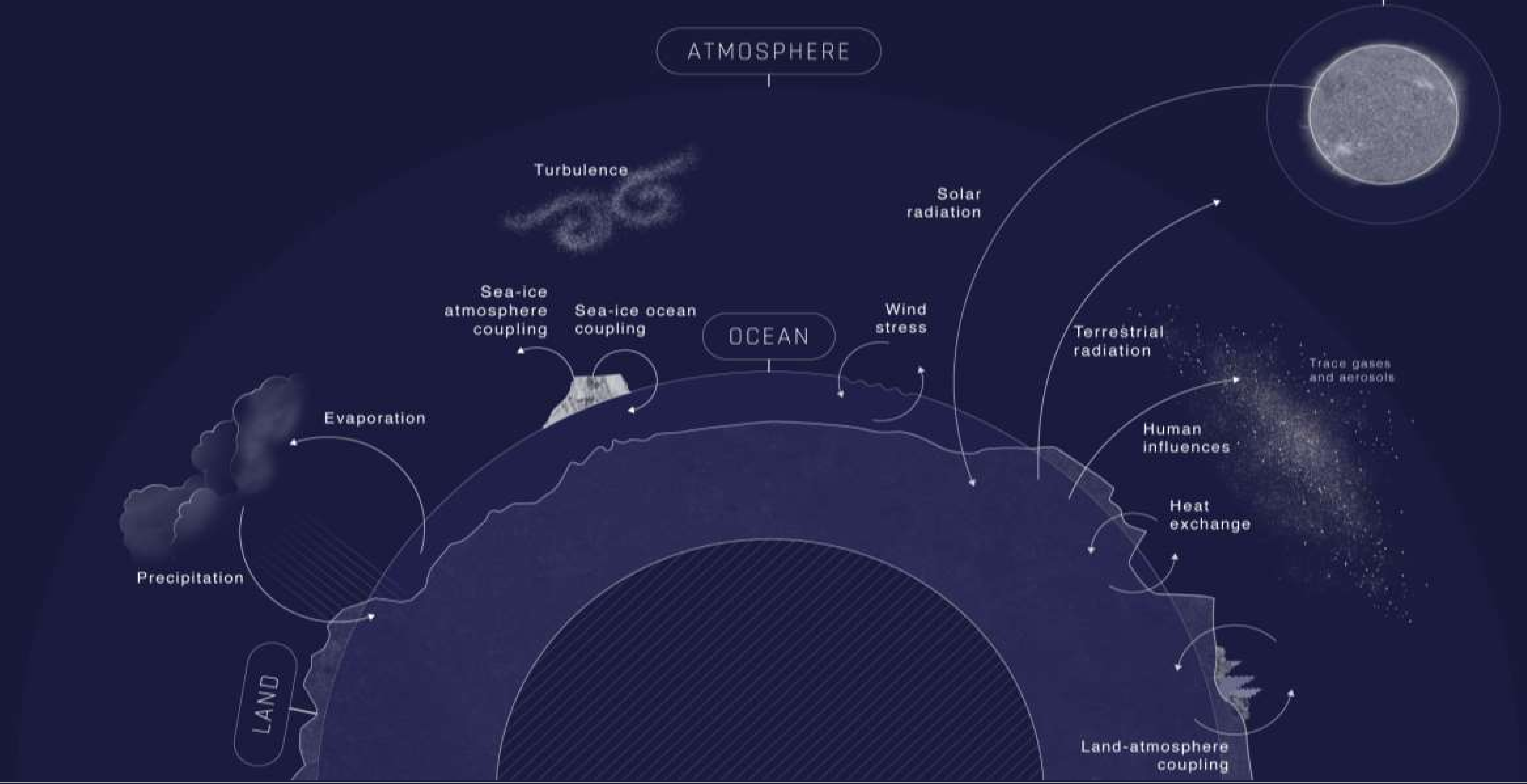
Modelling

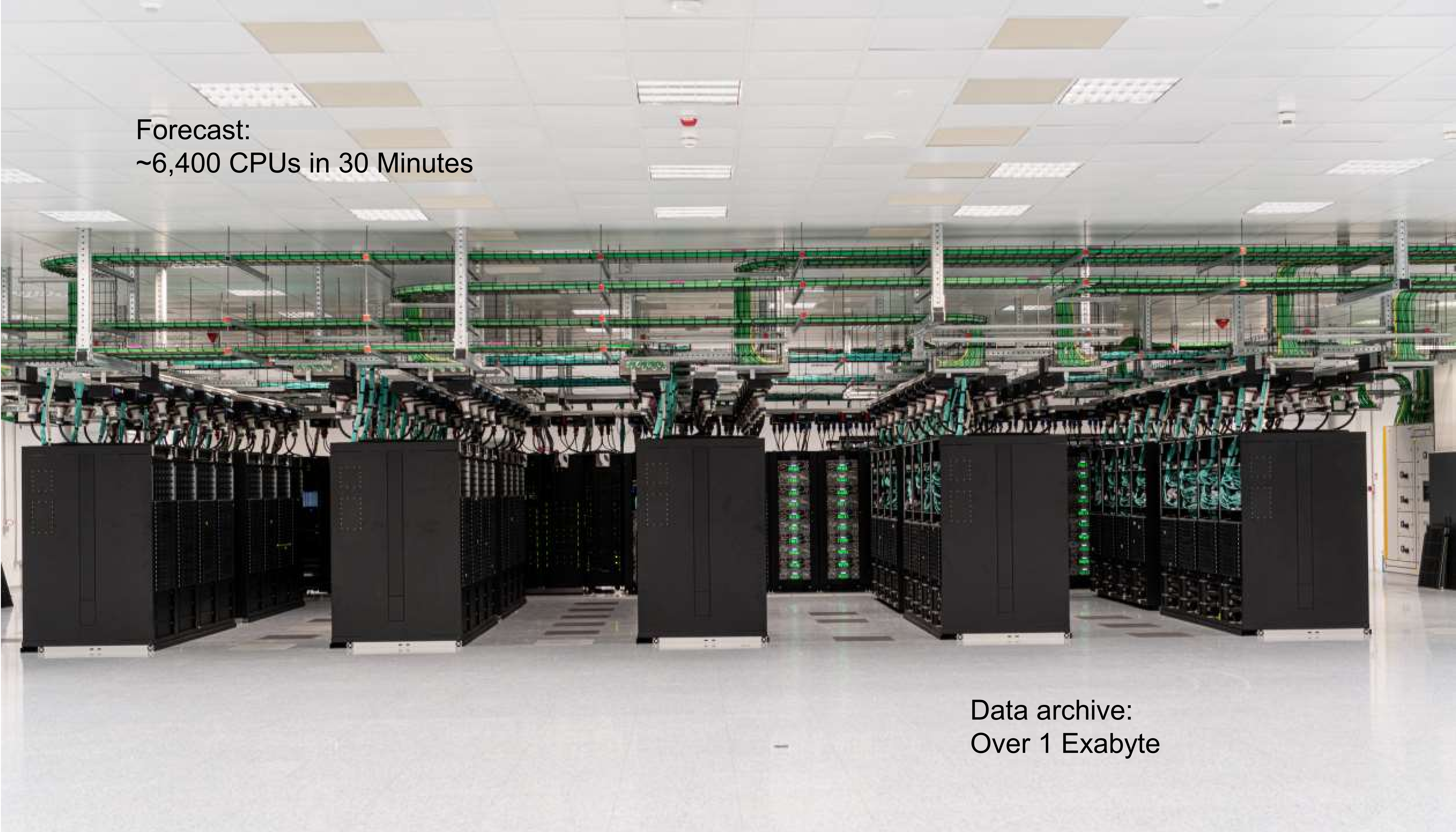
Predictions



800M
DAILY OBSERVATIONS

ECMWF EARTH SYSTEM APPROACH



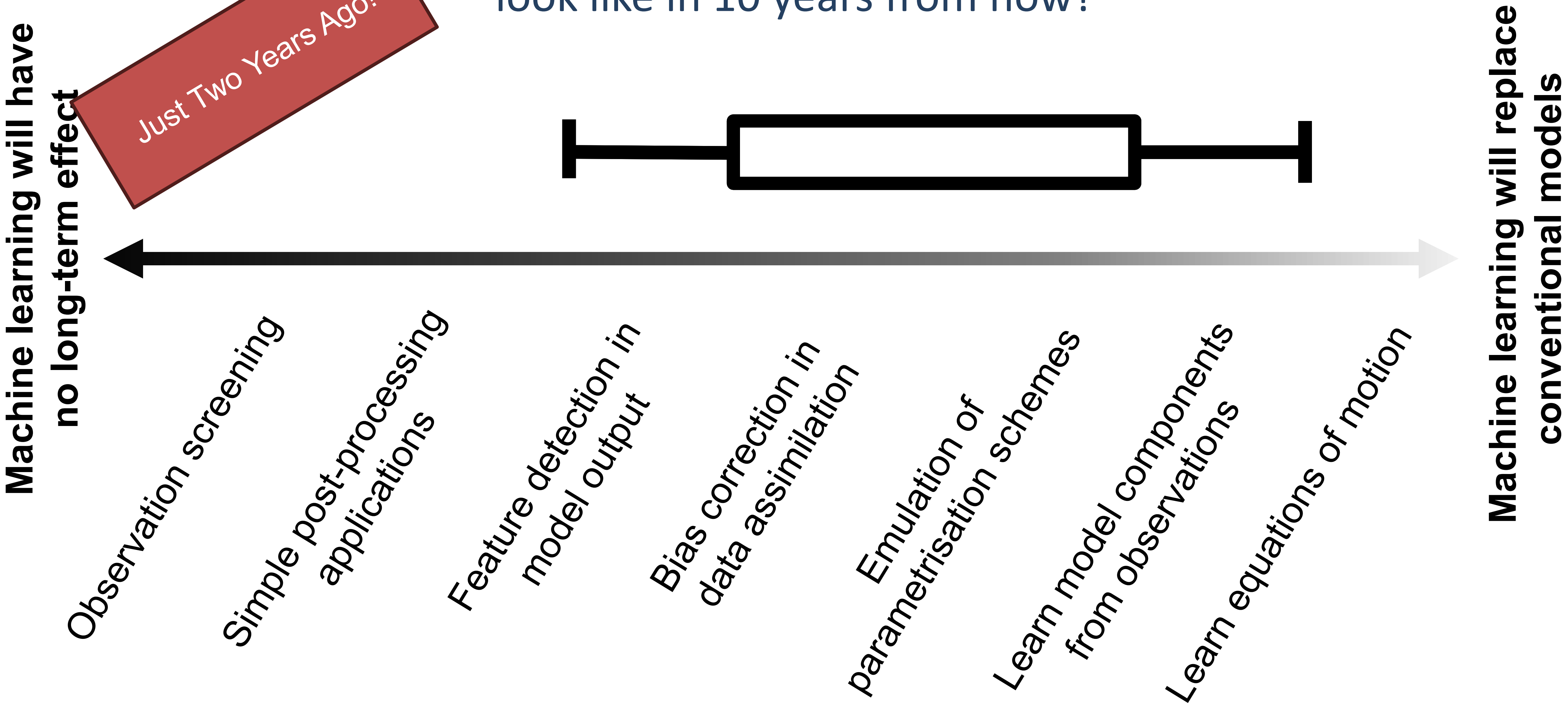


Forecast:
~6,400 CPUs in 30 Minutes

Data archive:
Over 1 Exabyte

What will machine learning for weather and climate predictions look like in 10 years from now?

Just Two Years Ago!



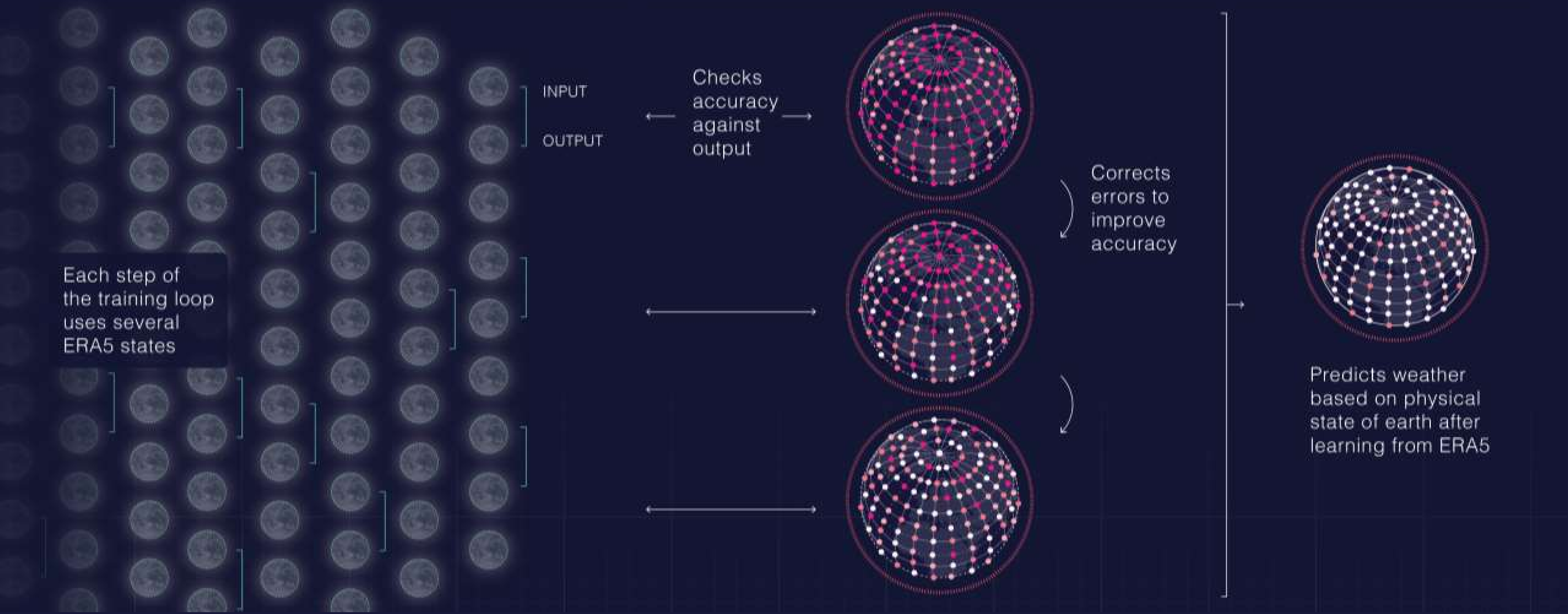
Machine learning will have no long-term effect
Machine learning will replace conventional models
Observation screening
Simple post-processing applications
Feature detection in model output
Bias correction in data assimilation
Emulation of parametrisation schemes
Learn model components from observations
Learn equations of motion

The uncertainty range is still very large...

Sets of training data from ERA5

Example of training loop

ML model

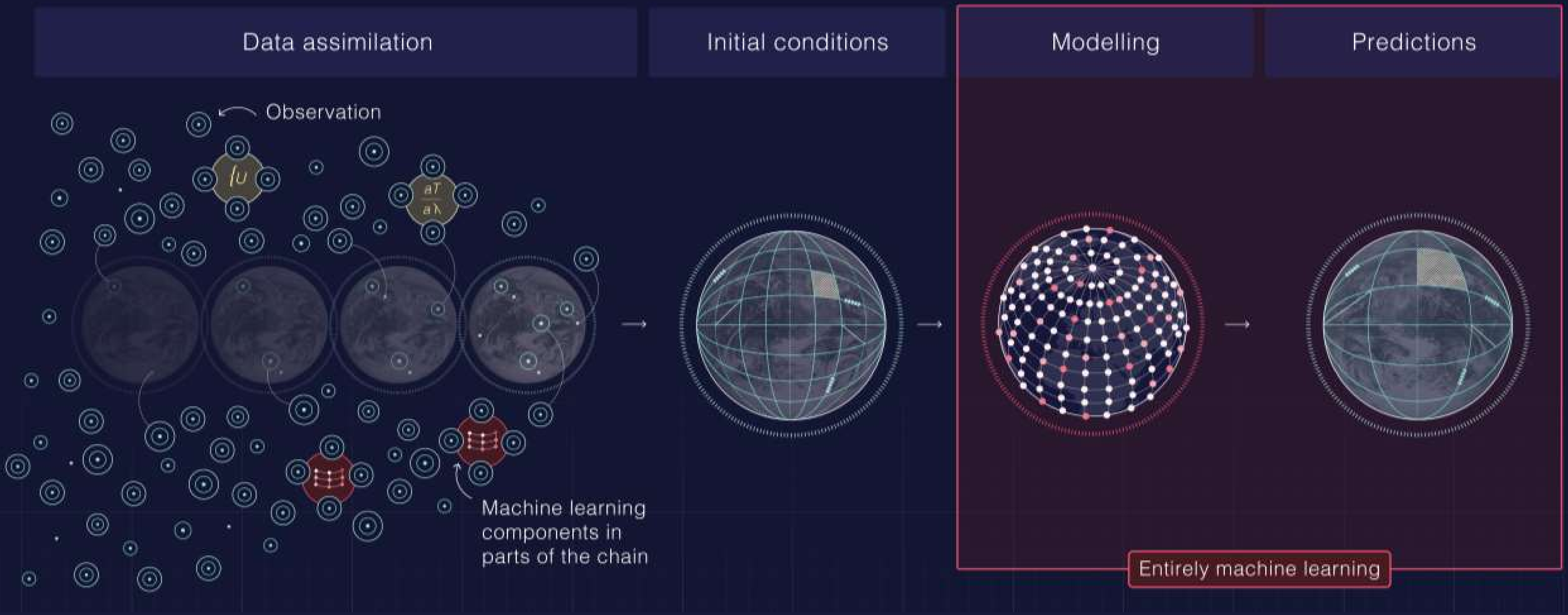


Each step of the training loop uses several ERA5 states

Checks accuracy against output

Corrects errors to improve accuracy

Predicts weather based on physical state of earth after learning from ERA5



Data assimilation

Initial conditions

Modelling

Predictions

Observation

Machine learning components in parts of the chain

Entirely machine learning

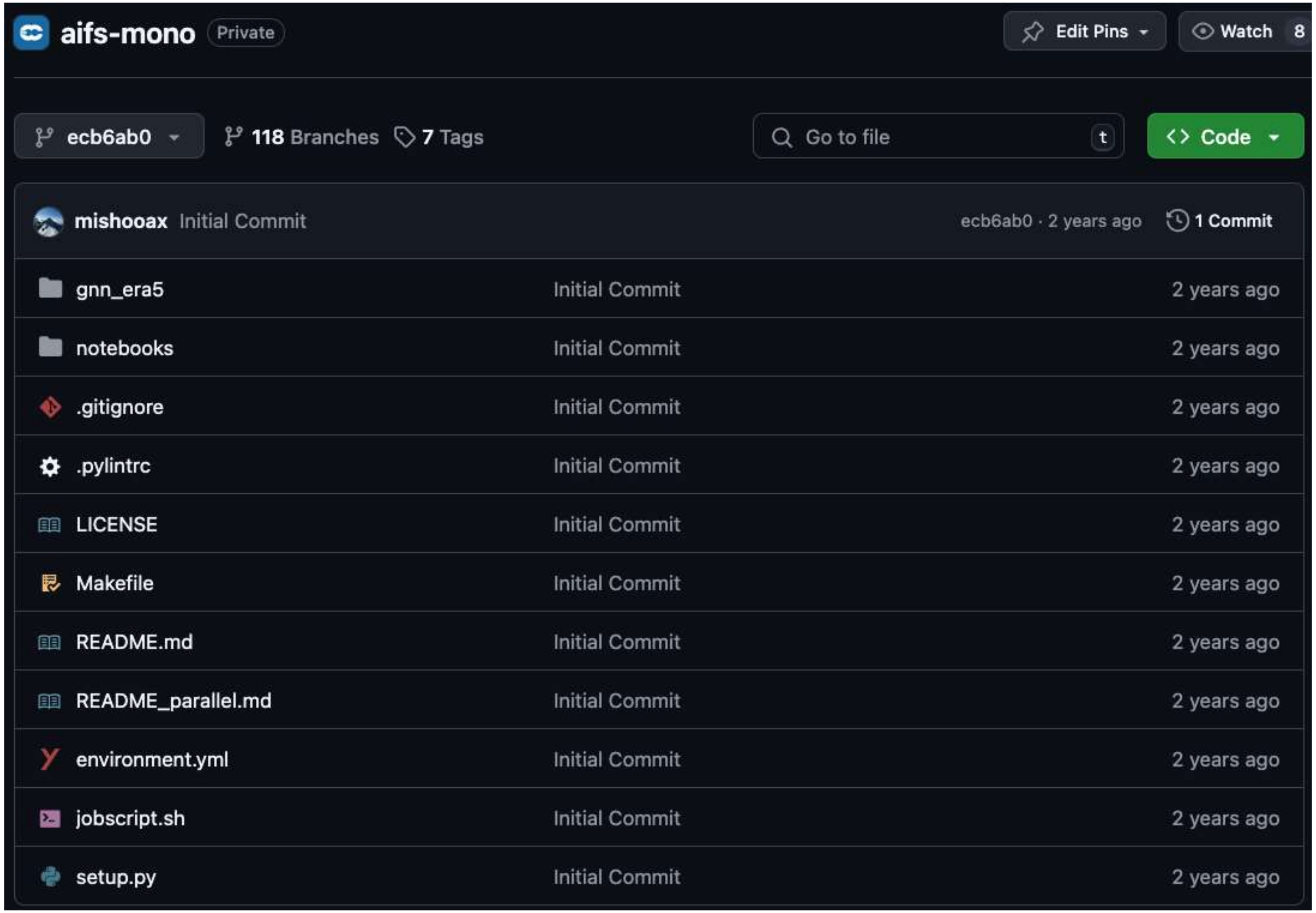
Training:
64 GPUs in 8 Days

Forecast:
1 GPU in 2 Minutes

ECMWF | EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

How did we end up here?

Initial Commit

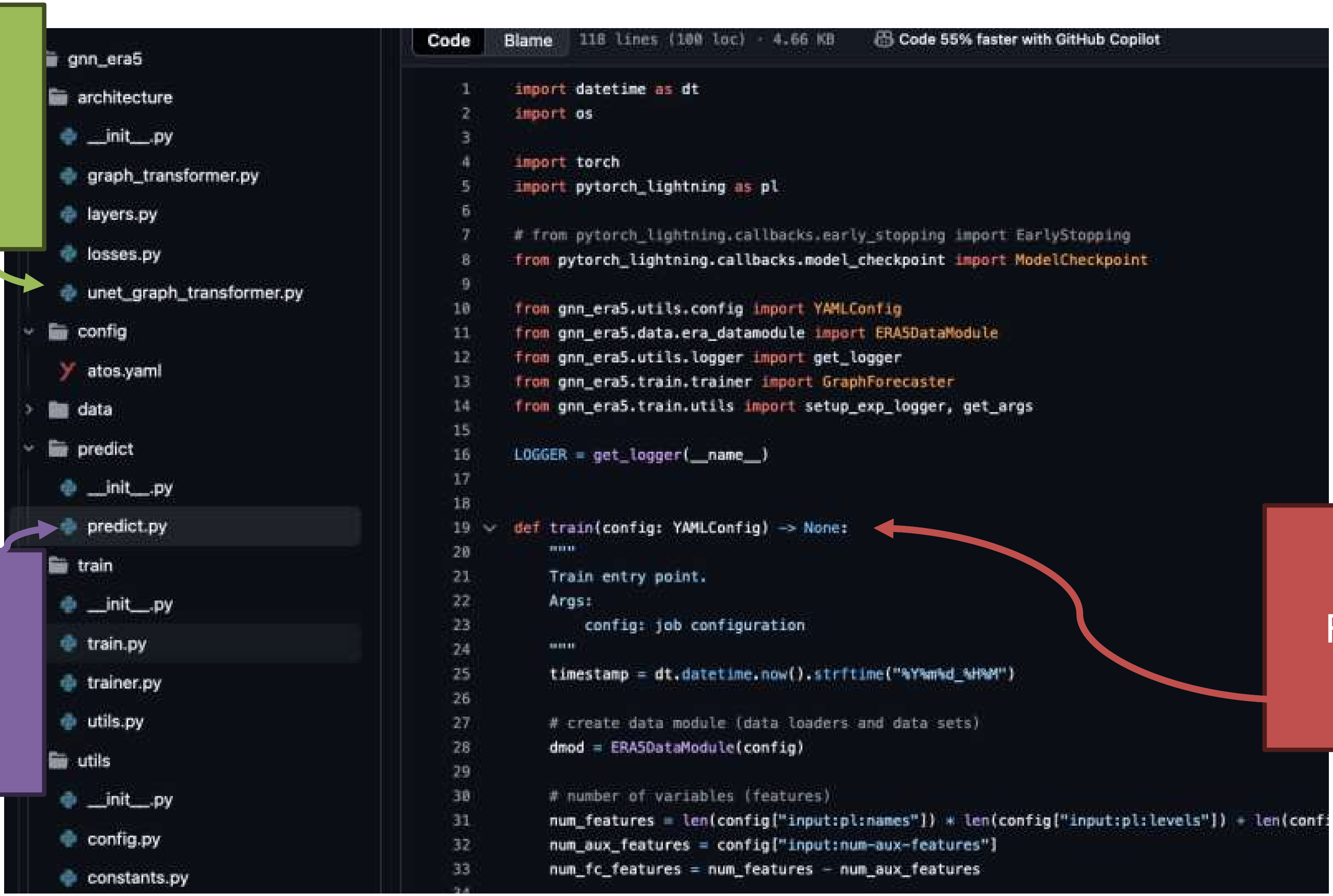


mishooax Initial Commit ecb6ab0 · 2 years ago 1 Commit

gnn_era5	Initial Commit	2 years ago
notebooks	Initial Commit	2 years ago
.gitignore	Initial Commit	2 years ago
.pylintrc	Initial Commit	2 years ago
LICENSE	Initial Commit	2 years ago
Makefile	Initial Commit	2 years ago
README.md	Initial Commit	2 years ago
README_parallel.md	Initial Commit	2 years ago
environment.yml	Initial Commit	2 years ago
jobscript.sh	Initial Commit	2 years ago
setup.py	Initial Commit	2 years ago

Typical research ML code

Trying out Architectures



```
gnn_era5
  architecture
    __init__.py
    graph_transformer.py
    layers.py
    losses.py
    unet_graph_transformer.py
  config
    atos.yaml
  data
  predict
    __init__.py
    predict.py
  train
    __init__.py
    train.py
    trainer.py
    utils.py
  utils
    __init__.py
    config.py
    constants.py
```

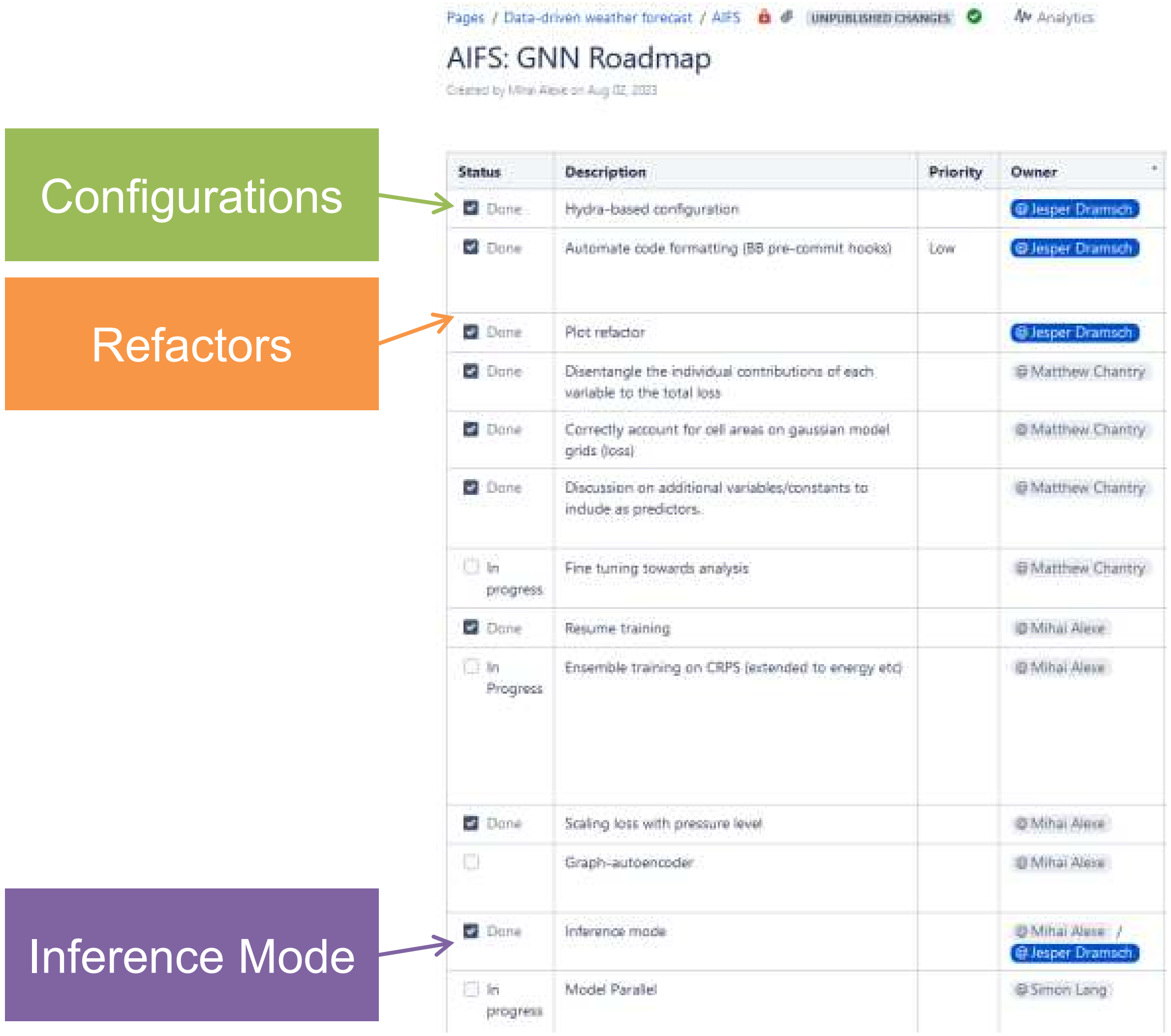
Code
Blame
118 lines (100 loc) · 4.66 KB
Code 55% faster with GitHub Copilot

```python
import datetime as dt
import os

import torch
import pytorch_lightning as pl

# from pytorch_lightning.callbacks.early_stopping import EarlyStopping
from pytorch_lightning.callbacks.model_checkpoint import ModelCheckpoint

from gnn_era5.utils.config import YAMLConfig
from gnn_era5.data.era_datamodule import ERA5DataModule
from gnn_era5.utils.logger import get_logger
from gnn_era5.train.trainer import GraphForecaster
from gnn_era5.train.utils import setup_exp_logger, get_args

LOGGER = get_logger(__name__)


def train(config: YAMLConfig) -> None:
    """
    Train entry point.
    Args:
        config: job configuration
    """
    timestamp = dt.datetime.now().strftime("%Y%m%d_%H%M")

    # create data module (data loaders and data sets)
    dmod = ERA5DataModule(config)

    # number of variables (features)
    num_features = len(config["input:pl:names"]) * len(config["input:pl:levels"]) + len(confi
    num_aux_features = config["input:num-aux-features"]
    num_fc_features = num_features - num_aux_features
```

Inference Code

Functional Code

It started out with a wish ✨



Configurations

Refactors

Inference Mode

Pages / Data-driven weather forecast / AIFS UNPUBLISHED CHANGES Analytics

AIFS: GNN Roadmap

Created by Mihai Alexe on Aug 02, 2023

Status	Description	Priority	Owner
☑ Done	Hydra-based configuration		@Jesper Dramsch
☑ Done	Automate code formatting (BB pre-commit hooks)	Low	@Jesper Dramsch
☑ Done	Plot refactor		@Jesper Dramsch
☑ Done	Disentangle the individual contributions of each variable to the total loss		@Matthew Chantry
☑ Done	Correctly account for cell areas on gaussian model grids (loss)		@Matthew Chantry
☑ Done	Discussion on additional variables/constants to include as predictors.		@Matthew Chantry
☐ In progress	Fine tuning towards analysis		@Matthew Chantry
☑ Done	Resume training		@Mihai Alexe
☐ In Progress	Ensemble training on CRPS (extended to energy etc)		@Mihai Alexe
☑ Done	Scaling loss with pressure level		@Mihai Alexe
☐	Graph-autoencoder		@Mihai Alexe
☑ Done	Inference mode		@Mihai Alexe / @Jesper Dramsch
☐ In progress	Model Parallel		@Simon Lang

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

Keeping the User and Collaboration in Mind

Researchers

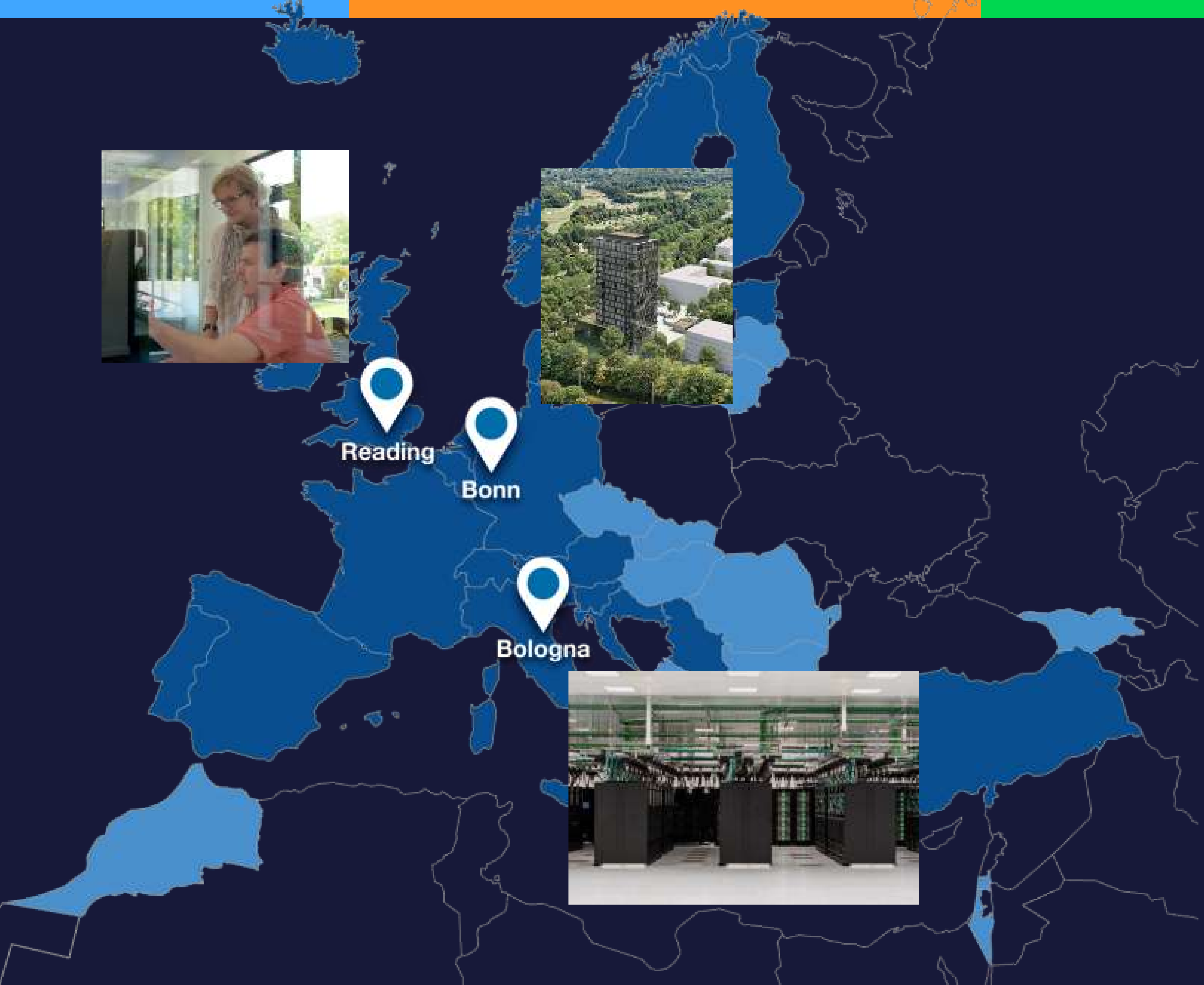
Developers

Operations

- Researchers
  - Quick switching of experiment values
  - Less interaction with core ML code
  - Low-key experiment tracking
- Developers
  - Modularity and Extensibility
  - Code quality
  - Separation of Concerns
- Operations
  - Minimal Dependencies
  - Consistent interfaces

How do we facilitate collaboration?

ECMWF | EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

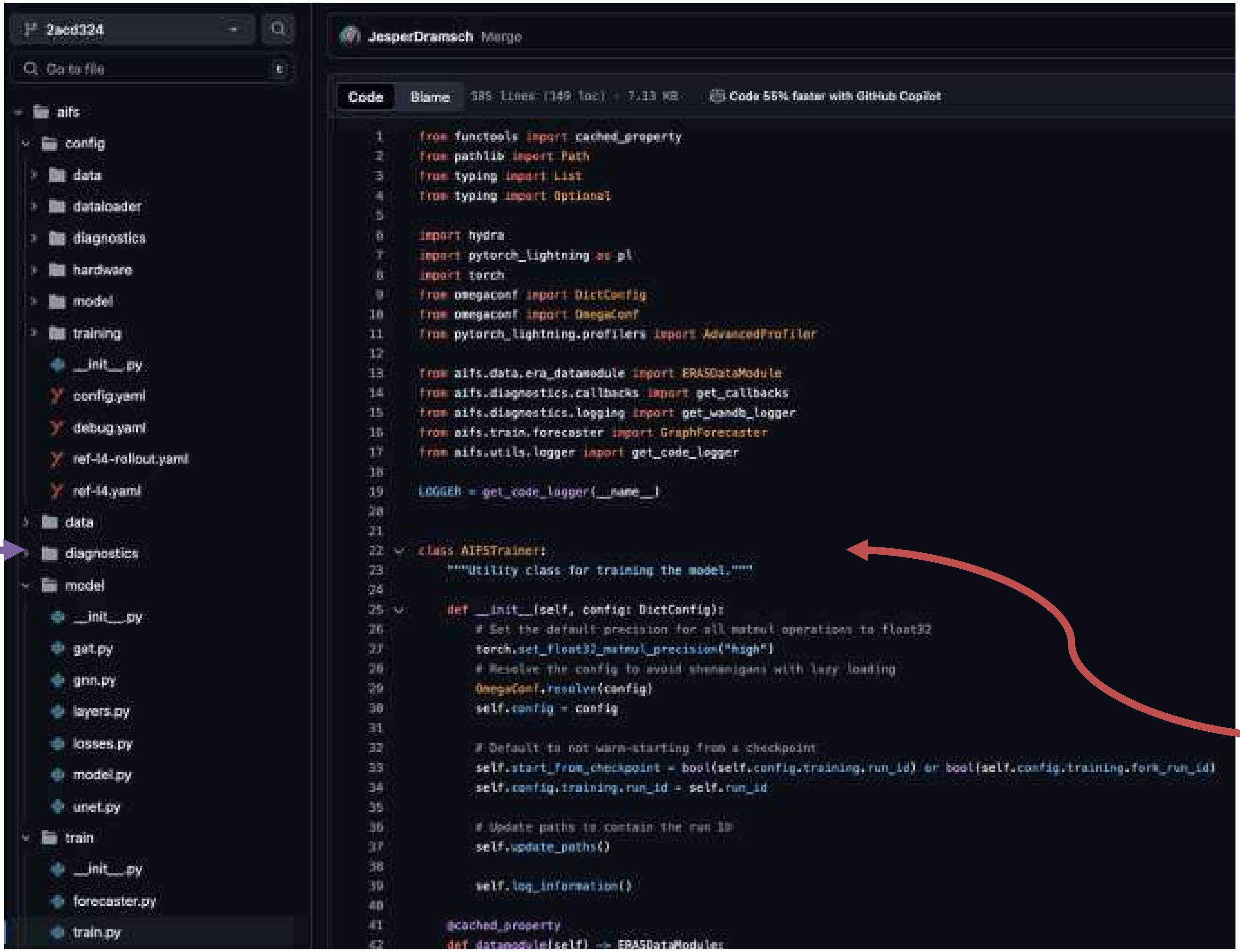

Reading
Bonn
Bologna

Modular and Extensible AIFS Trainer code

Hierarchical Configs with Hydra

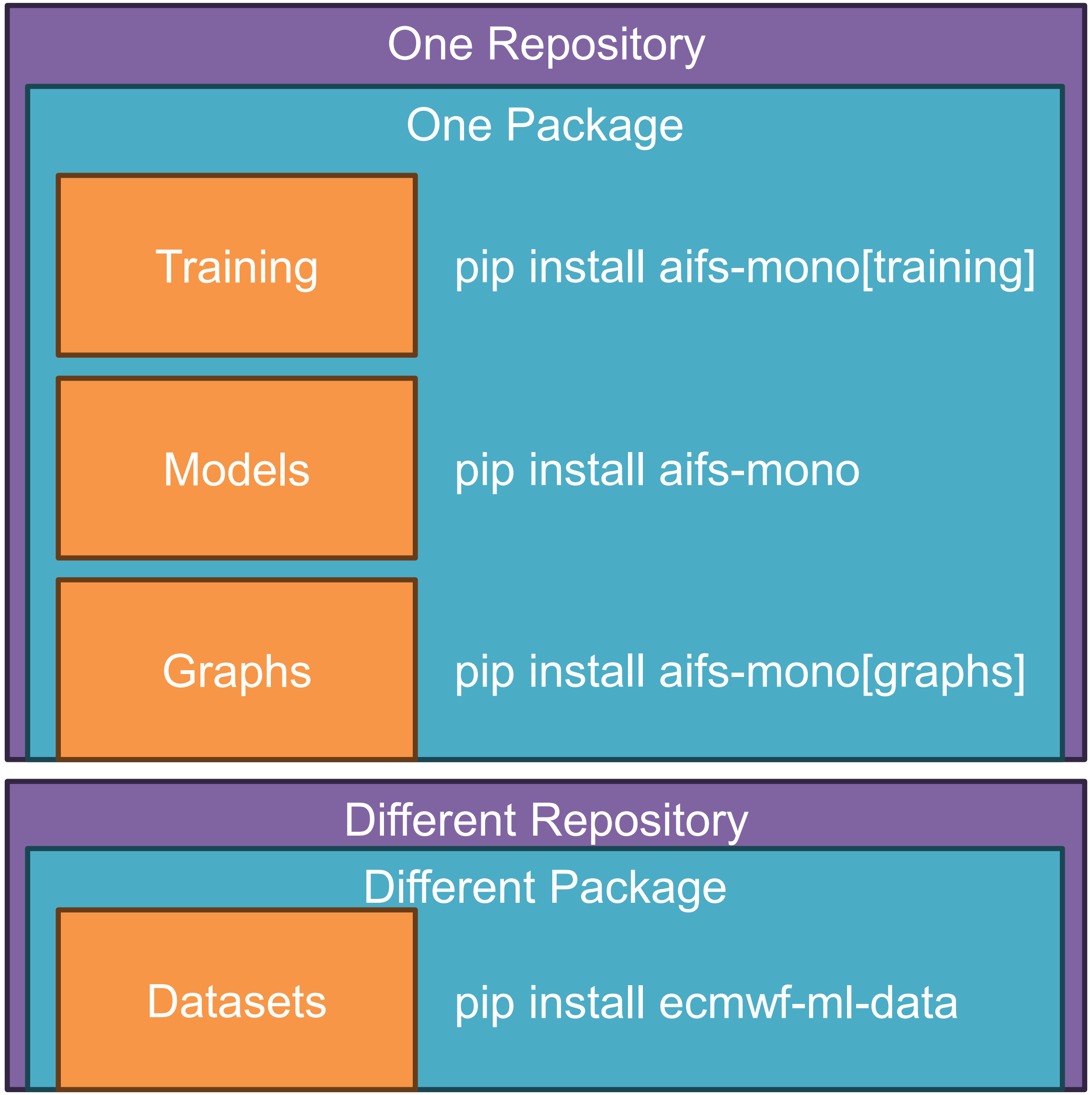
Tidier Separation

Modular Trainer Class



```python
from functools import cached_property
from pathlib import Path
from typing import List
from typing import Optional

import hydra
import pytorch_lightning as pl
import torch
from omegaconf import DictConfig
from omegaconf import OmegaConf
from pytorch_lightning.profilers import AdvancedProfiler

from aifs.data.era_datamodule import ERA5DataModule
from aifs.diagnostics.callbacks import get_callbacks
from aifs.diagnostics.logging import get_wandb_logger
from aifs.train.forecaster import GraphForecaster
from aifs.utils.logger import get_code_logger

LOGGER = get_code_logger(__name__)


class AIFSTrainer:
    """Utility class for training the model."""

    def __init__(self, config: DictConfig):
        # Set the default precision for all matmul operations to float32
        torch.set_float32_matmul_precision("high")
        # Resolve the config to avoid shenanigans with lazy loading
        OmegaConf.resolve(config)
        self.config = config

        # Default to not warm-starting from a checkpoint
        self.start_from_checkpoint = bool(self.config.training.run_id) or bool(self.config.training.fork_run_id)
        self.config.training.run_id = self.run_id

        # Update paths to contain the run ID
        self.update_paths()

        self.log_information()

    @cached_property
    def datamodule(self) -> ERA5DataModule:
```

AIFS mono-package + External dependencies



One Repository

One Package

Training — pip install aifs-mono[training]

Models — pip install aifs-mono

Graphs — pip install aifs-mono[graphs]

Different Repository

Different Package

Datasets — pip install ecmwf-ml-data

- Pros
  - ▲ Convenient to develop
  - ▲ All code in one place
  - ▲ Quick to release

- Cons
  - ▼ Weird to install
  - ▼ Not all code in „ecosystem“
  - ▼ Complex
  - ▼ No unified testing or infrastructure

Focusing on Configurability, Extensibility and Modularity

```
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.
AnemoiModelEncProcDec

processor:
  _target_:
anemoi.models.layers.processor.GraphTransformerPr
ocessor
  _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

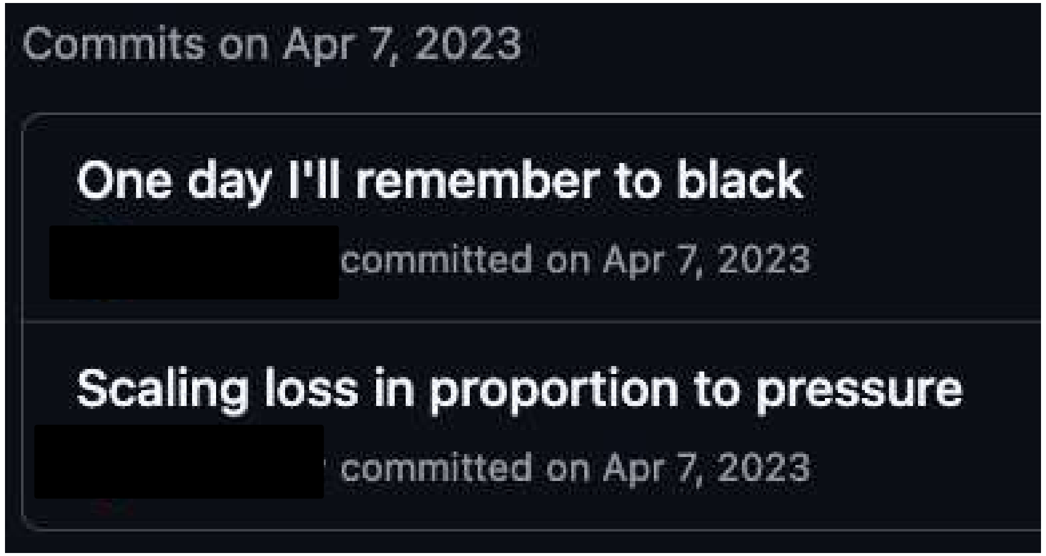
```
Model / Transformer.yml
activation: GELU
num_channels: 1024

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

processor:
  _target_:
anemoi.models.layers.processor.TransformerProcesso
r
  _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 with hydra
- Full tracking even on terminal overrides
- Easy to extend with new models and components

refactored Simon‘s new transformer
experiment tracking with Ana and Sara

Why we use pre-commit hooks



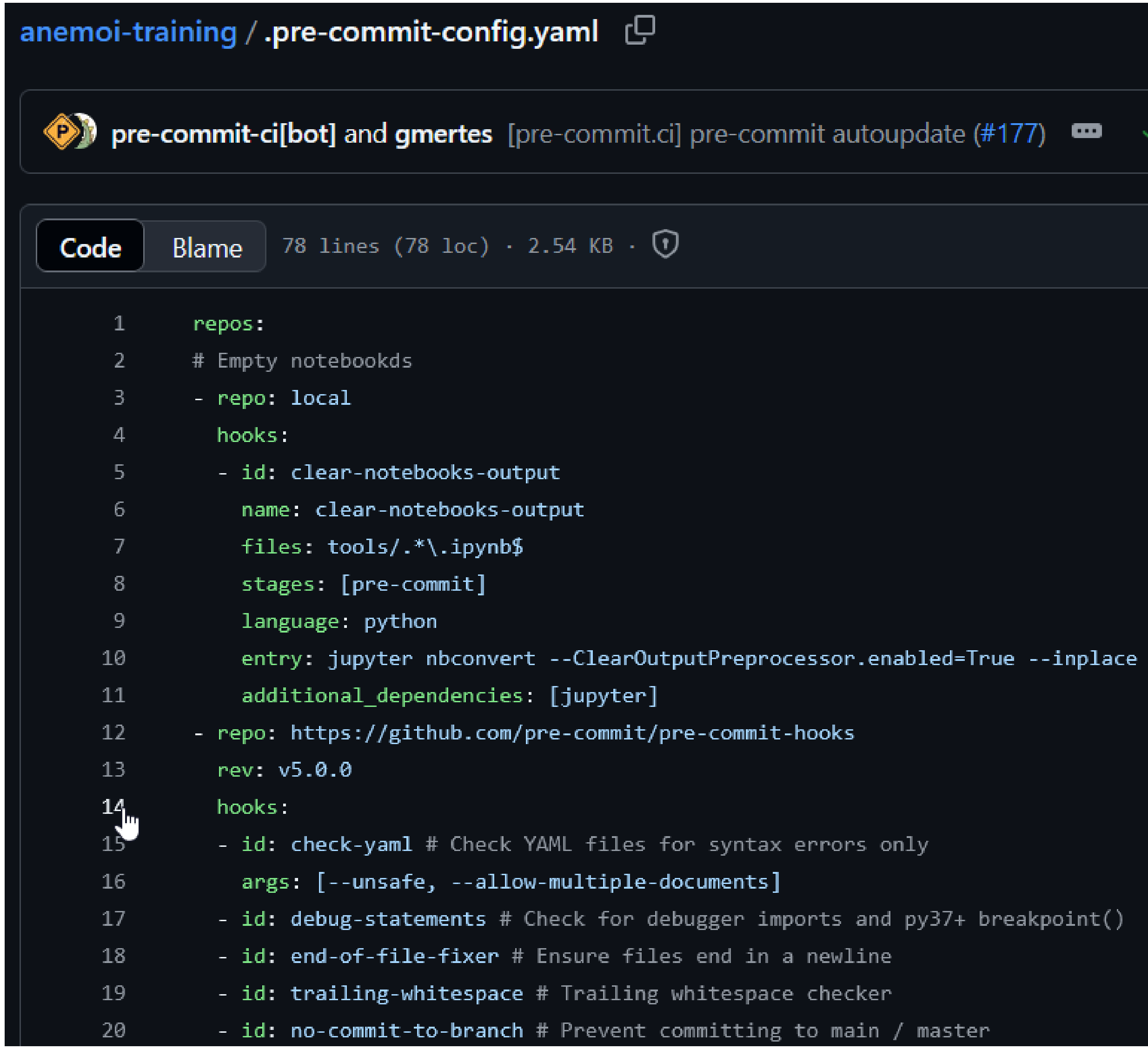
Commits on Apr 7, 2023

One day I'll remember to black

[redacted] committed on Apr 7, 2023

Scaling loss in proportion to pressure

[redacted] committed on Apr 7, 2023



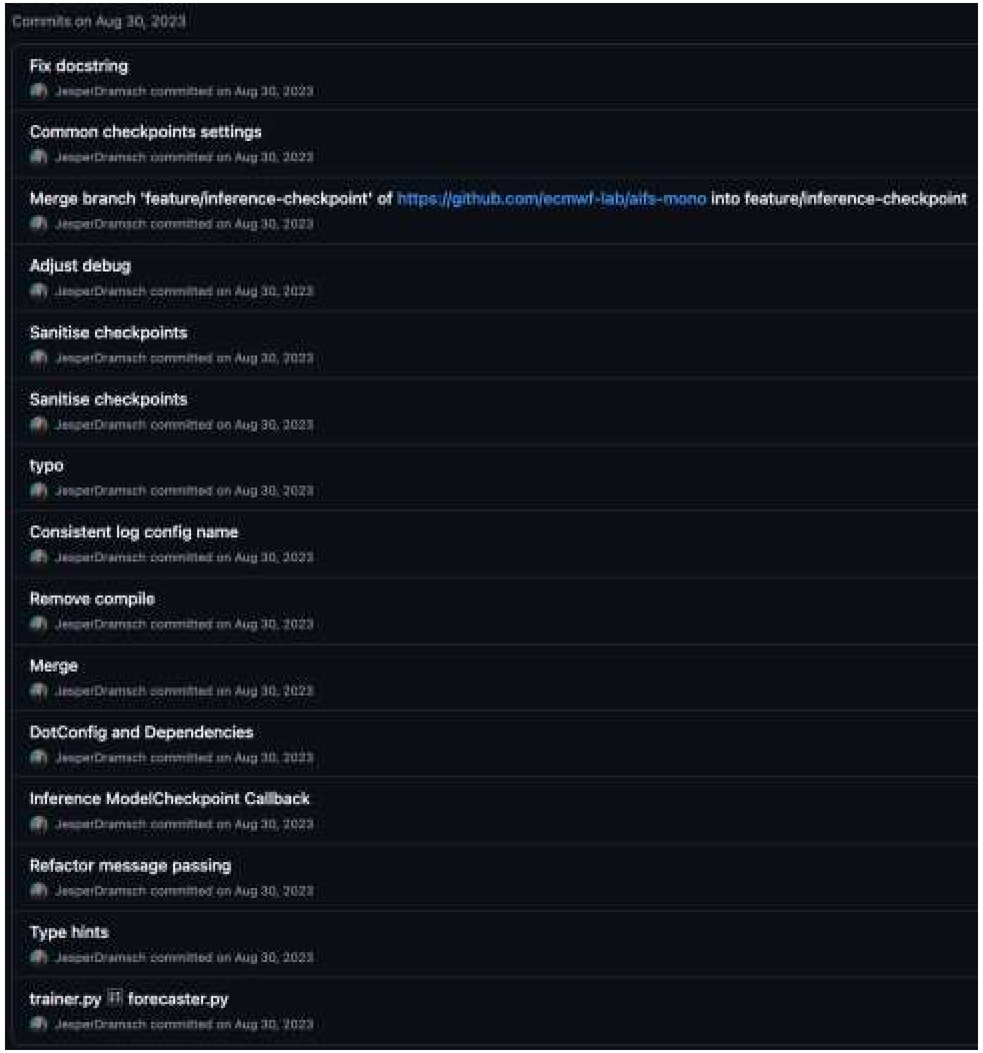
anemoi-training / .pre-commit-config.yaml

pre-commit-ci[bot] and gmertes [pre-commit.ci] pre-commit autoupdate (#177)

78 lines (78 loc) · 2.54 KB ·

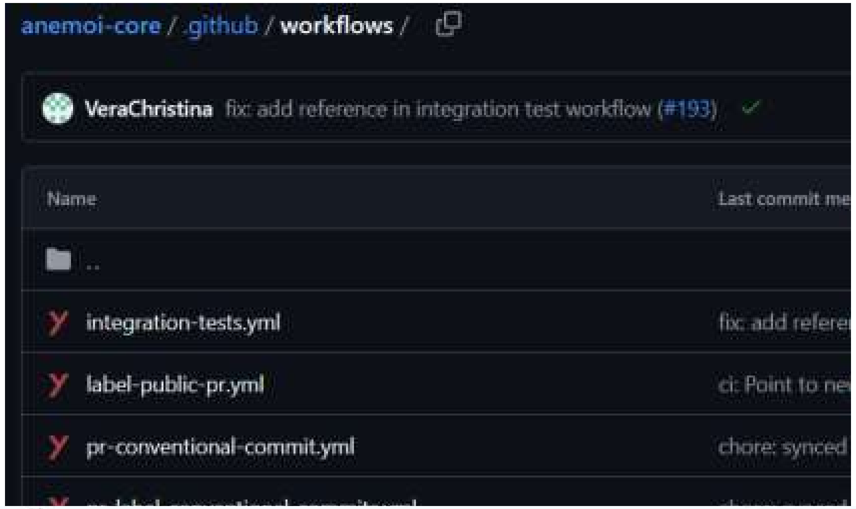
```yaml
repos:
# Empty notebookds
- repo: local
  hooks:
  - id: clear-notebooks-output
    name: clear-notebooks-output
    files: tools/.*\.ipynb$
    stages: [pre-commit]
    language: python
    entry: jupyter nbconvert --ClearOutputPreprocessor.enabled=True --inplace
    additional_dependencies: [jupyter]
- repo: https://github.com/pre-commit/pre-commit-hooks
  rev: v5.0.0
  hooks:
  - id: check-yaml # Check YAML files for syntax errors only
    args: [--unsafe, --allow-multiple-documents]
  - id: debug-statements # Check for debugger imports and py37+ breakpoint()
  - id: end-of-file-fixer # Ensure files end in a newline
  - id: trailing-whitespace # Trailing whitespace checker
  - id: no-commit-to-branch # Prevent committing to main / master
```

Why we do squash commits now 👀



Commits on Aug 30, 2023

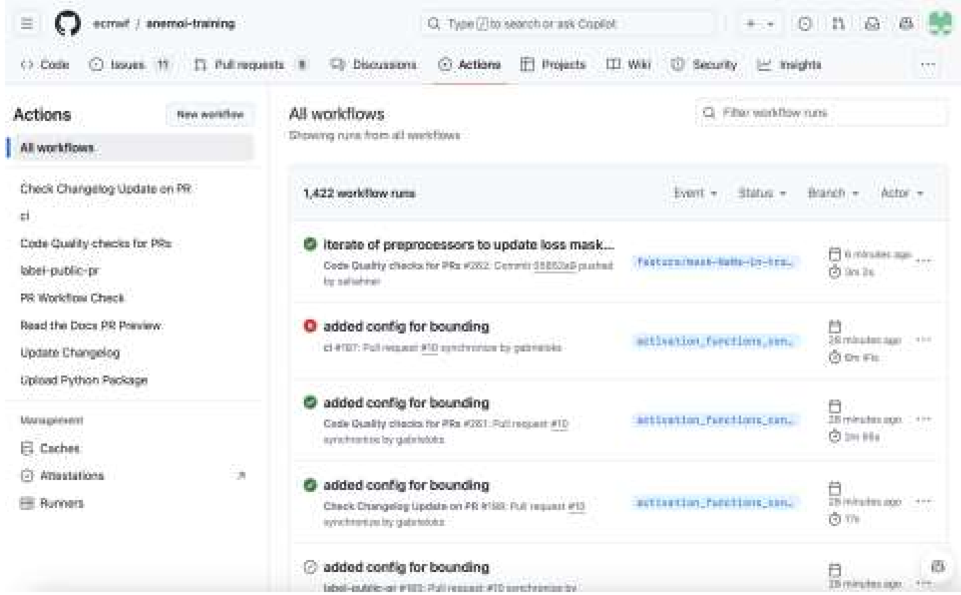
- **Fix docstring**
  JesperDramsch committed on Aug 30, 2023
- **Common checkpoints settings**
  JesperDramsch committed on Aug 30, 2023
- **Merge branch 'feature/inference-checkpoint' of https://github.com/ecmwf-lab/aifs-mono into feature/inference-checkpoint**
  JesperDramsch committed on Aug 30, 2023
- **Adjust debug**
  JesperDramsch committed on Aug 30, 2023
- **Sanitise checkpoints**
  JesperDramsch committed on Aug 30, 2023
- **Sanitise checkpoints**
  JesperDramsch committed on Aug 30, 2023
- **typo**
  JesperDramsch committed on Aug 30, 2023
- **Consistent log config name**
  JesperDramsch committed on Aug 30, 2023
- **Remove compile**
  JesperDramsch committed on Aug 30, 2023
- **Merge**
  JesperDramsch committed on Aug 30, 2023
- **DotConfig and Dependencies**
  JesperDramsch committed on Aug 30, 2023
- **Inference ModelCheckpoint Callback**
  JesperDramsch committed on Aug 30, 2023
- **Refactor message passing**
  JesperDramsch committed on Aug 30, 2023
- **Type hints**
  JesperDramsch committed on Aug 30, 2023
- **trainer.py ⇆ forecaster.py**
  JesperDramsch committed on Aug 30, 2023



anemoi-core / .github / workflows /

VeraChristina fix: add reference in integration test workflow (#193) ✓

Name	Last commit me
..	
integration-tests.yml	fix: add refere
label-public-pr.yml	ci: Point to ne
pr-conventional-commit.yml	chore: synced



ecmwf / anemoi-training

Actions

- All workflows
- Check Changelog Update on PR
- ci
- Code Quality checks for PRs
- label-public-pr
- PR Workflow Check
- Read the Docs PR Preview
- Update Changelog
- Upload Python Package

Management

- Caches
- Attestations
- Runners

All workflows

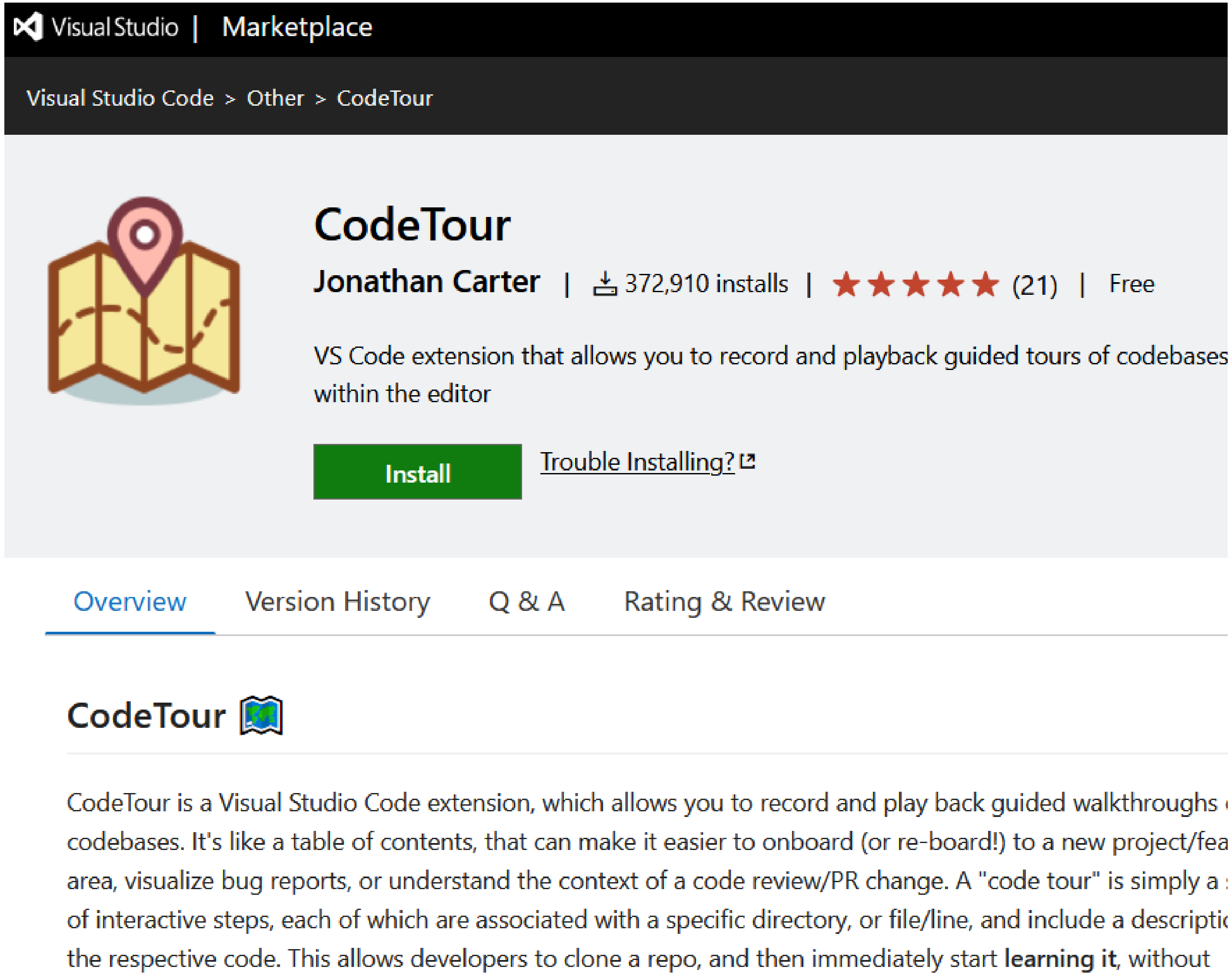
Showing runs from all workflows

1,422 workflow runs

- ✓ **Iterate of preprocessors to update loss mask...**
  Code Quality checks for PRs #262: Commit 05852a0 pushed by sahahner
  feature/mask-fields-in-tra...
  6 minutes ago
  3m 2s
- ✗ **added config for bounding**
  ci #187: Pull request #10 synchronize by gabrieloks
  activation_functions_con...
  28 minutes ago
  0m 41s
- ✓ **added config for bounding**
  Code Quality checks for PRs #261: Pull request #10 synchronize by gabrieloks
  activation_functions_con...
  28 minutes ago
  3m 46s
- ✓ **added config for bounding**
  Check Changelog Update on PR #188: Pull request #10 synchronize by gabrieloks
  activation_functions_con...
  28 minutes ago
  17s
- **added config for bounding**
  label-public-pr #183: Pull request #10 synchronize by
  28 minutes ago

Do Code Tours!

- Shortcuts learning (multi-week -> 1 hour)
- Puts design decisions in context
- Gives New Hires space to ask questions
- Can be tailored to specific needs
- Establishes collaborative aspects
- Optionally: use tools



CodeTour

Jonathan Carter | 372,910 installs | ★★★★★ (21) | Free

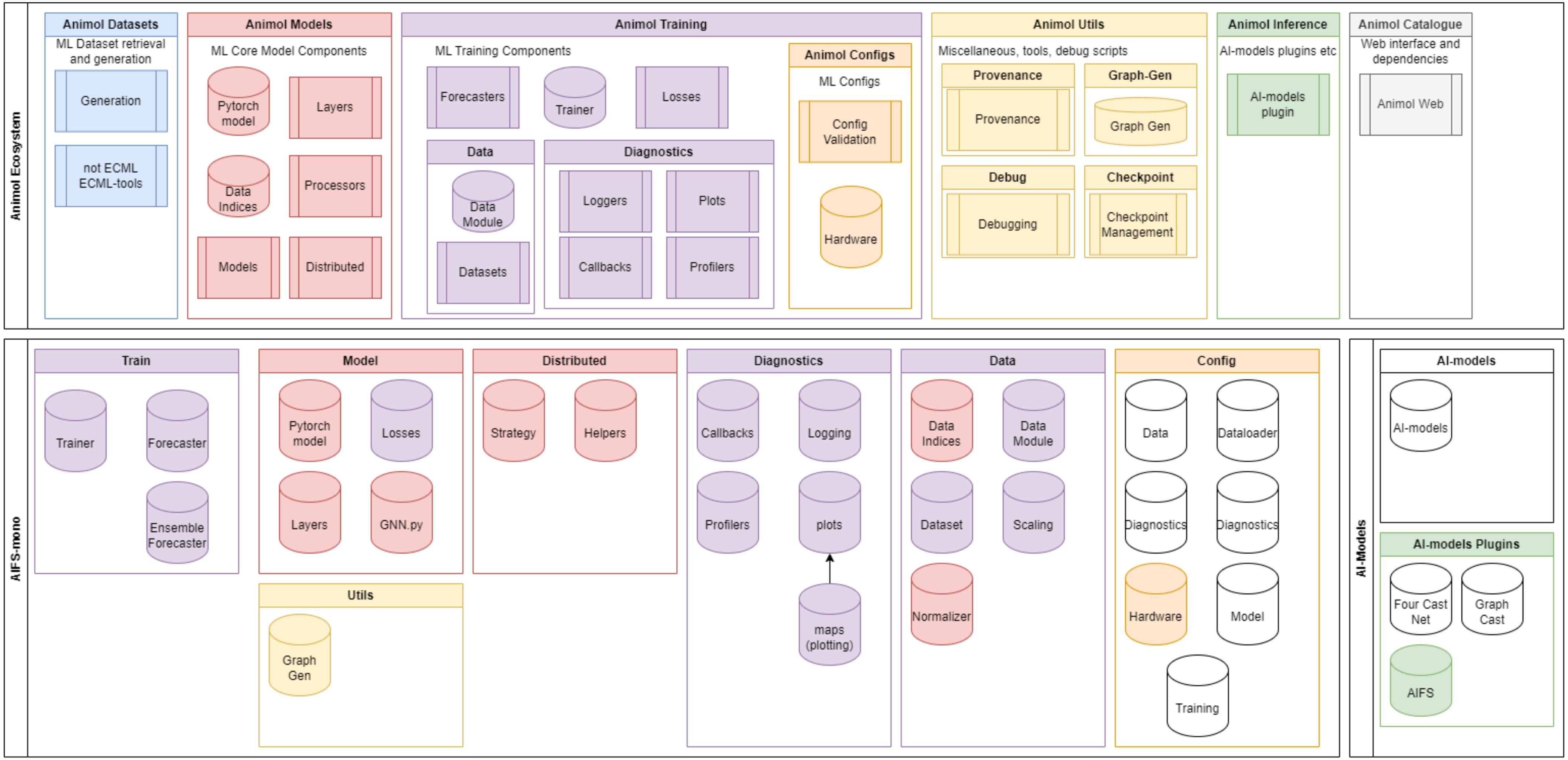
VS Code extension that allows you to record and playback guided tours of codebases within the editor

CodeTour

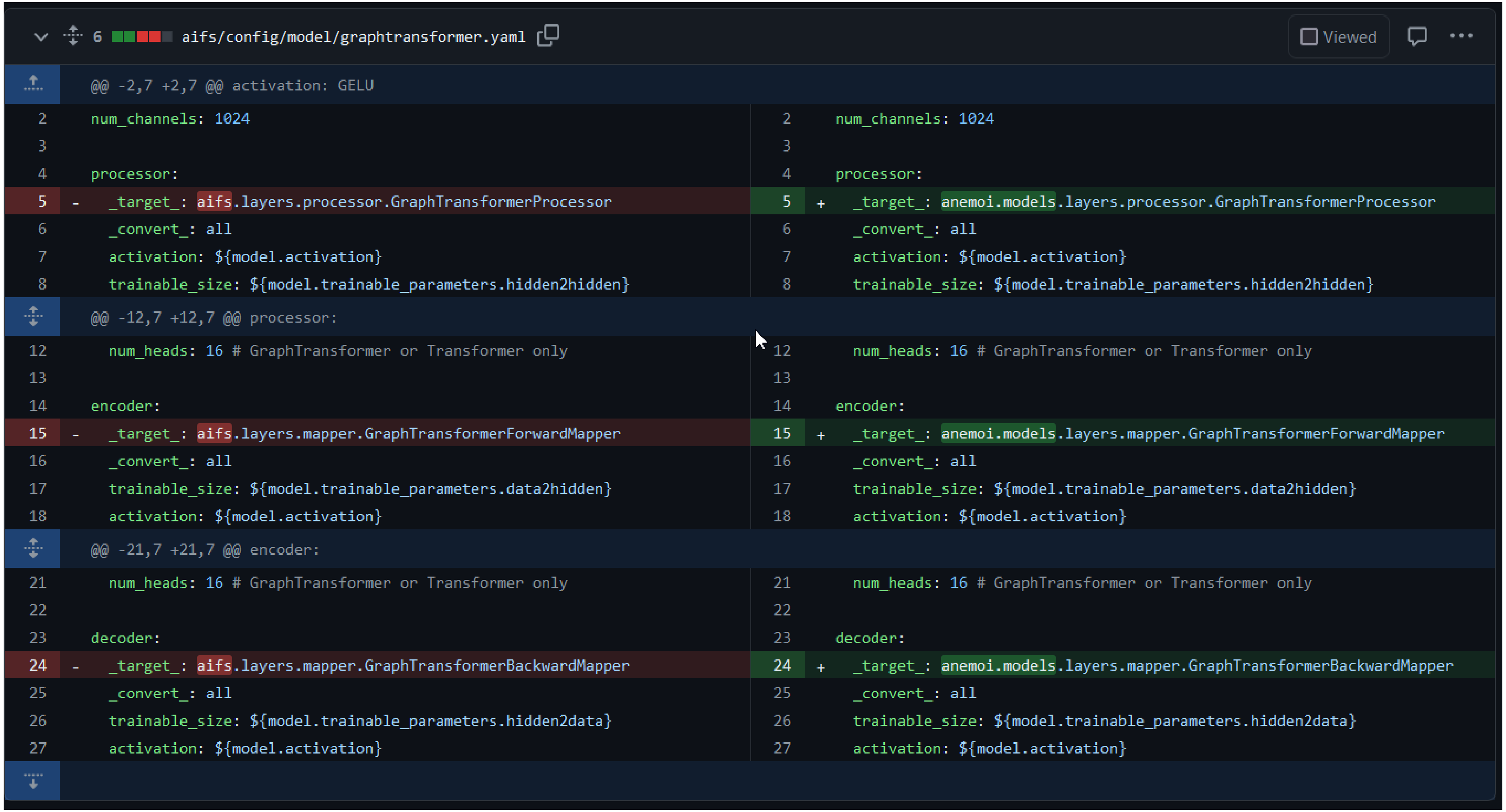
CodeTour is a Visual Studio Code extension, which allows you to record and play back guided walkthroughs codebases. It's like a table of contents, that can make it easier to onboard (or re-board!) to a new project/fea area, visualize bug reports, or understand the context of a code review/PR change. A "code tour" is simply a of interactive steps, each of which are associated with a specific directory, or file/line, and include a descriptio the respective code. This allows developers to clone a repo, and then immediately start **learning it**, without

Growing beyond a single git repo

The Vision of AIFS becoming the Anemoi Ecosystem

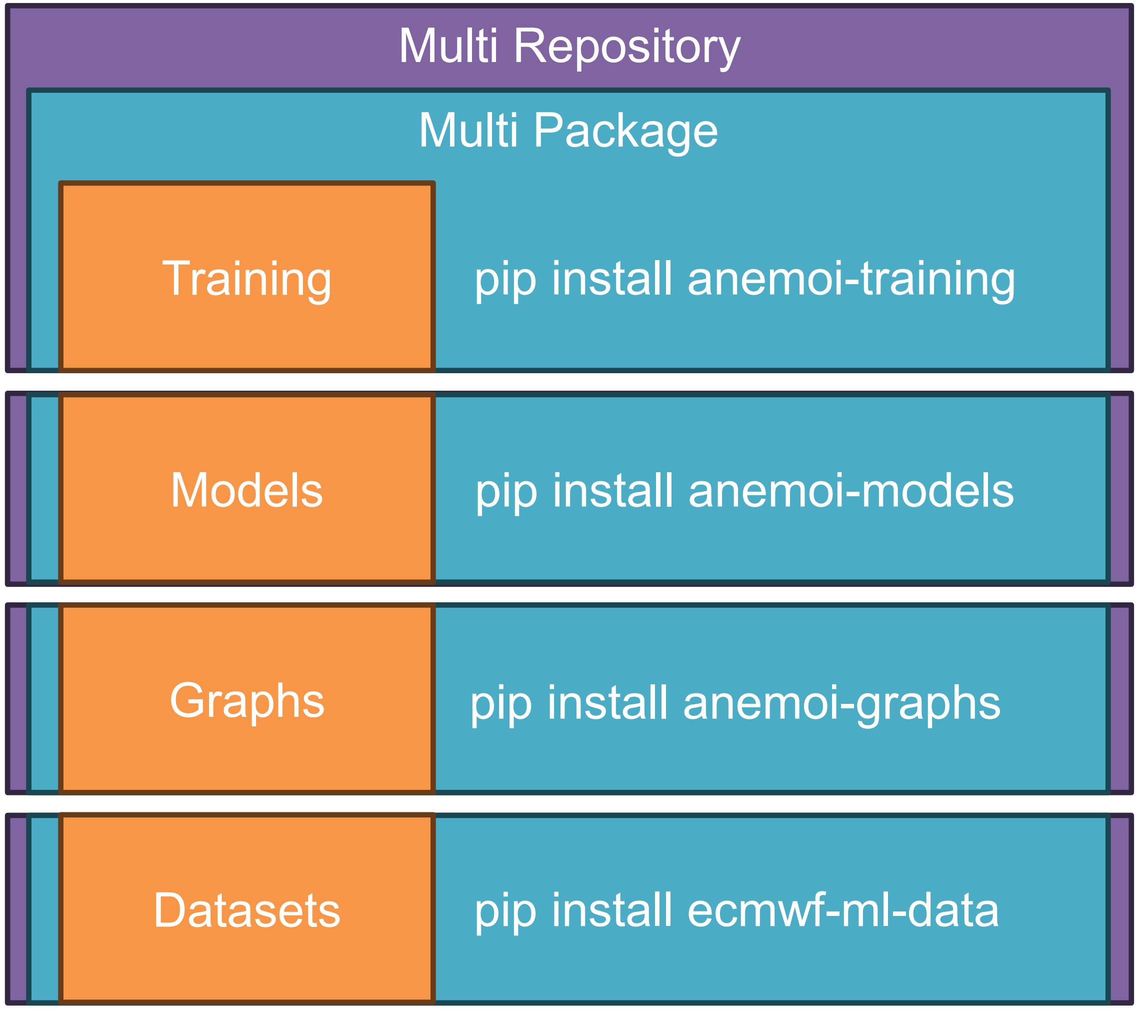


Common interface and modularity



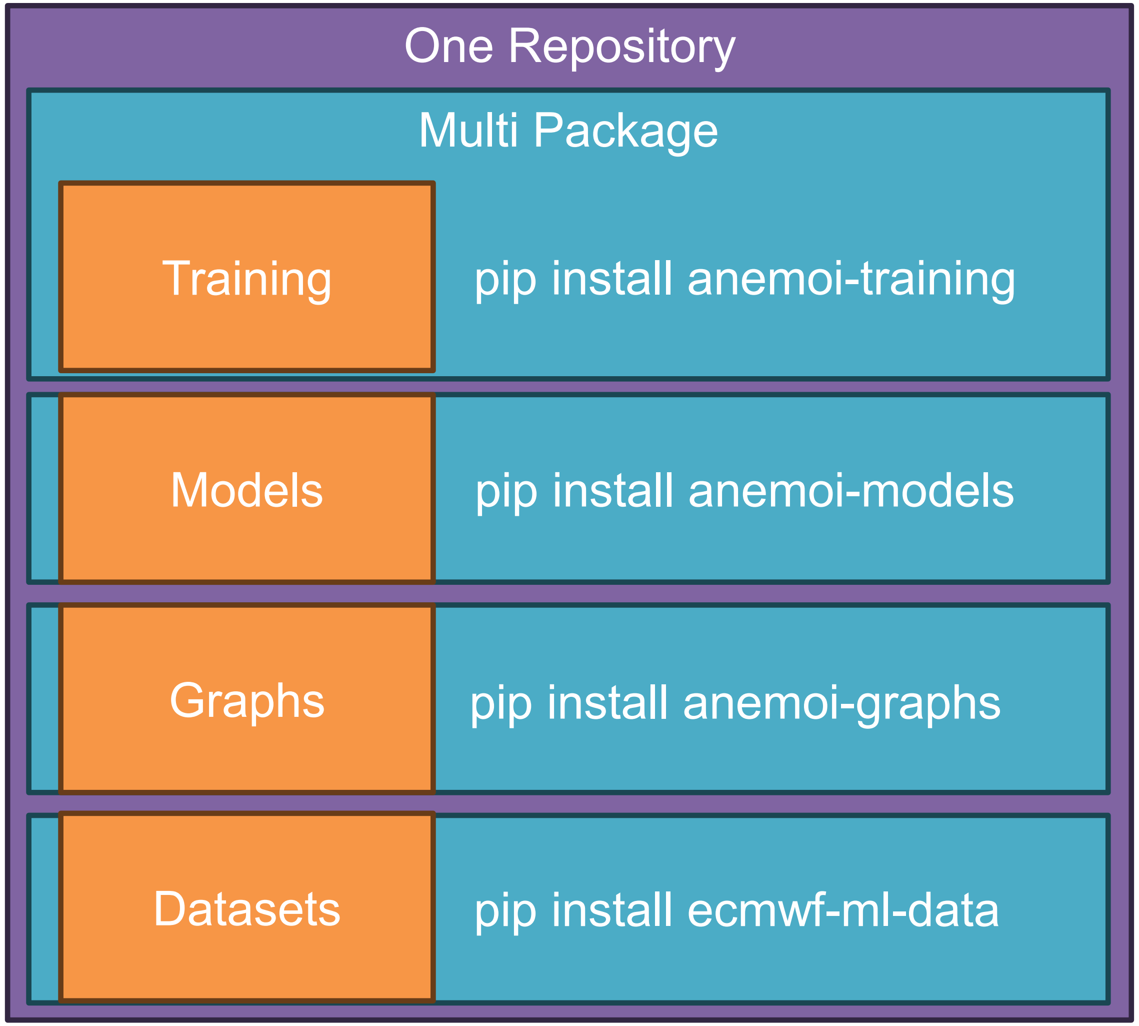
```
6 aifs/config/model/graphtransformer.yaml
@@ -2,7 +2,7 @@ activation: GELU
 num_channels: 1024

 processor:
-  _target_: aifs.layers.processor.GraphTransformerProcessor
+  _target_: anemoi.models.layers.processor.GraphTransformerProcessor
   _convert_: all
   activation: ${model.activation}
   trainable_size: ${model.trainable_parameters.hidden2hidden}
@@ -12,7 +12,7 @@ processor:
   num_heads: 16 # GraphTransformer or Transformer only

 encoder:
-  _target_: aifs.layers.mapper.GraphTransformerForwardMapper
+  _target_: anemoi.models.layers.mapper.GraphTransformerForwardMapper
   _convert_: all
   trainable_size: ${model.trainable_parameters.data2hidden}
   activation: ${model.activation}
@@ -21,7 +21,7 @@ encoder:
   num_heads: 16 # GraphTransformer or Transformer only

 decoder:
-  _target_: aifs.layers.mapper.GraphTransformerBackwardMapper
+  _target_: anemoi.models.layers.mapper.GraphTransformerBackwardMapper
   _convert_: all
   trainable_size: ${model.trainable_parameters.hidden2data}
   activation: ${model.activation}
```

Anemoi Ecosystem (multi repo)


Multi Repository
Multi Package
Training
pip install anemoi-training
Models
pip install anemoi-models
Graphs
pip install anemoi-graphs
Datasets
pip install ecmwf-ml-data

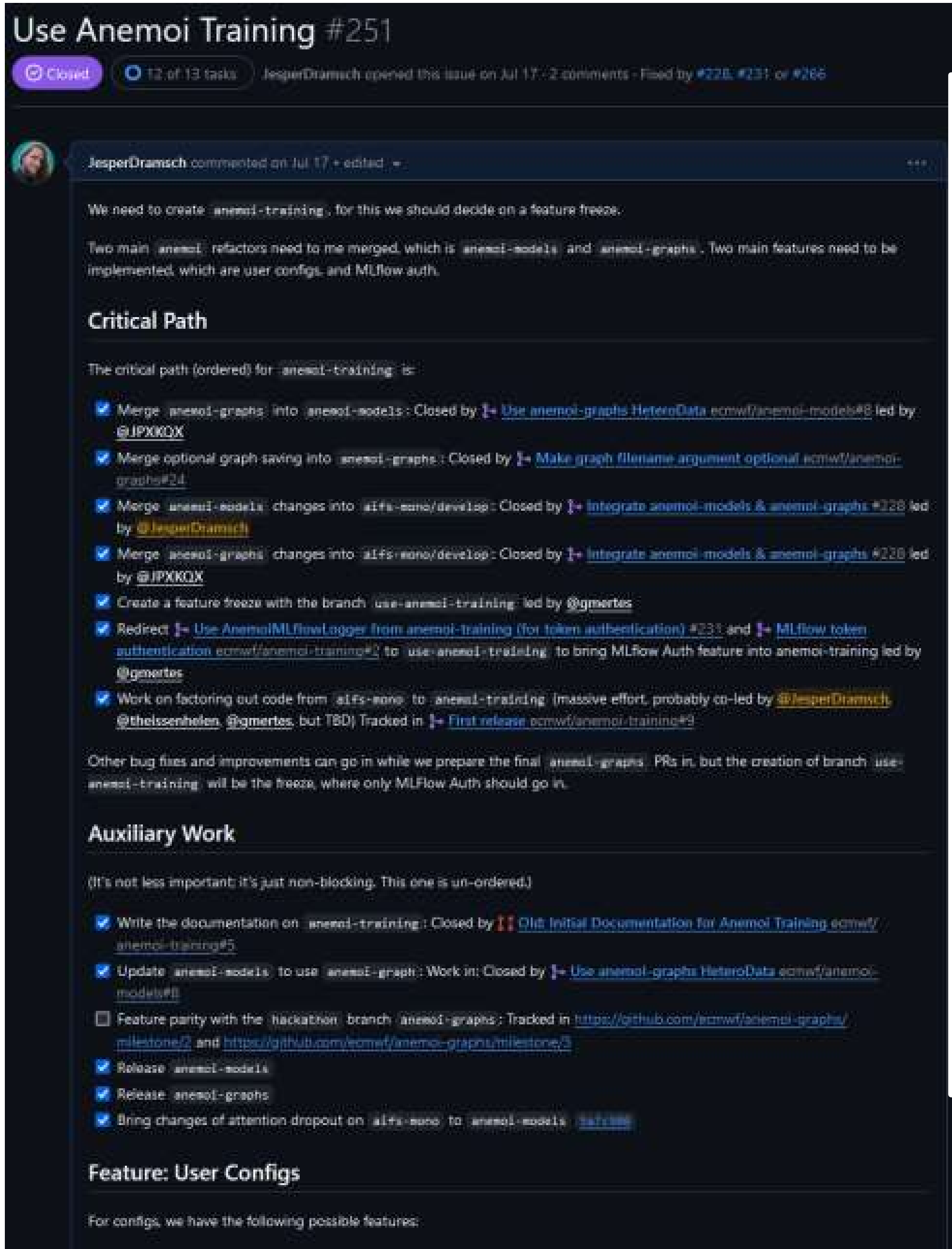
- Pros
  - ▲ Separation of concerns
  - ▲ Neat repo structure at root
  - ▲ Intuitive to Release individually
  - ▲ Complexity contained within repo
  - ▲ Easy to delegate responsibility
  - ▲ Different styles of collaboration possible
- Cons
  - ▼ PRs across repos for changes
    - ▼ Working on „develop“ can break
    - ▼ Out of sync
    - ▼ Synced Releases difficult
  - ▼ CI/CD hard to set up and maintain
  - ▼ Dependencies are hell to manage
  - ▼ End-to-end tests difficult to set up
  - ▼ Could silo knowledge in teams/repos
  - ▼ Can be difficult „keeping up“

Alternative: Anemoi Ecosystem (mono repo)


One Repository
Multi Package
Training
pip install anemoi-training
Models
pip install anemoi-models
Graphs
pip install anemoi-graphs
Datasets
pip install ecmwf-ml-data

- Pros
  - ▲ Single PR for change across packages
    - ▲ Easier Refactors too
  - ▲ Easy CI including end-to-end tests
  - ▲ Easiest security scanning
  - ▲ Consistent coding standards
  - ▲ Common state of all „main"s
- Cons
  - ▼ „main" will move even faster
  - ▼ Complex release cycle
  - ▼ Different working styles might clash
  - ▼ Large repository size can make git slow
  - ▼ Might need „mono-repo tools"
  - ▼ Branching strategy complex
  - ▼ Easy to break „everything" accidentally

Releasing Anemoi Training to the World



Use Anemoi Training #251

JesperDramsch opened this issue on Jul 17 · 2 comments · Fixed by #228, #231 or #266

JesperDramsch commented on Jul 17 • edited

We need to create `anemoi-training`, for this we should decide on a feature freeze.

Two main `anemoi` refactors need to me merged, which is `anemoi-models` and `anemoi-graphs`. Two main features need to be implemented, which are user configs, and MLflow auth.

Critical Path

The critical path (ordered) for `anemoi-training` is:

- ☑ Merge `anemoi-graphs` into `anemoi-models`: Closed by Use anemoi-graphs HeteroData ecmwf/anemoi-models#8 led by @JPXKQX
- ☑ Merge optional graph saving into `anemoi-graphs`: Closed by Make graph filename argument optional ecmwf/anemoi-graphs#24
- ☑ Merge `anemoi-models` changes into `aifs-mono/develop`: Closed by Integrate anemoi-models & anemoi-graphs #228 led by @JesperDramsch
- ☑ Merge `anemoi-graphs` changes into `aifs-mono/develop`: Closed by Integrate anemoi-models & anemoi-graphs #228 led by @JPXKQX
- ☑ Create a feature freeze with the branch `use-anemoi-training` led by @gmertes
- ☑ Redirect Use AnemoiMLflowLogger from anemoi-training (for token authentication) #231 and MLflow token authentication ecmwf/anemoi-training#2 to `use-anemoi-training` to bring MLflow Auth feature into anemoi-training led by @gmertes
- ☑ Work on factoring out code from `aifs-mono` to `anemoi-training` (massive effort, probably co-led by @JesperDramsch, @theissenhelen, @gmertes, but TBD) Tracked in First release ecmwf/anemoi-training#9

Other bug fixes and improvements can go in while we prepare the final `anemoi-graphs` PRs in, but the creation of branch `use-anemoi-training` will be the freeze, where only MLFlow Auth should go in.

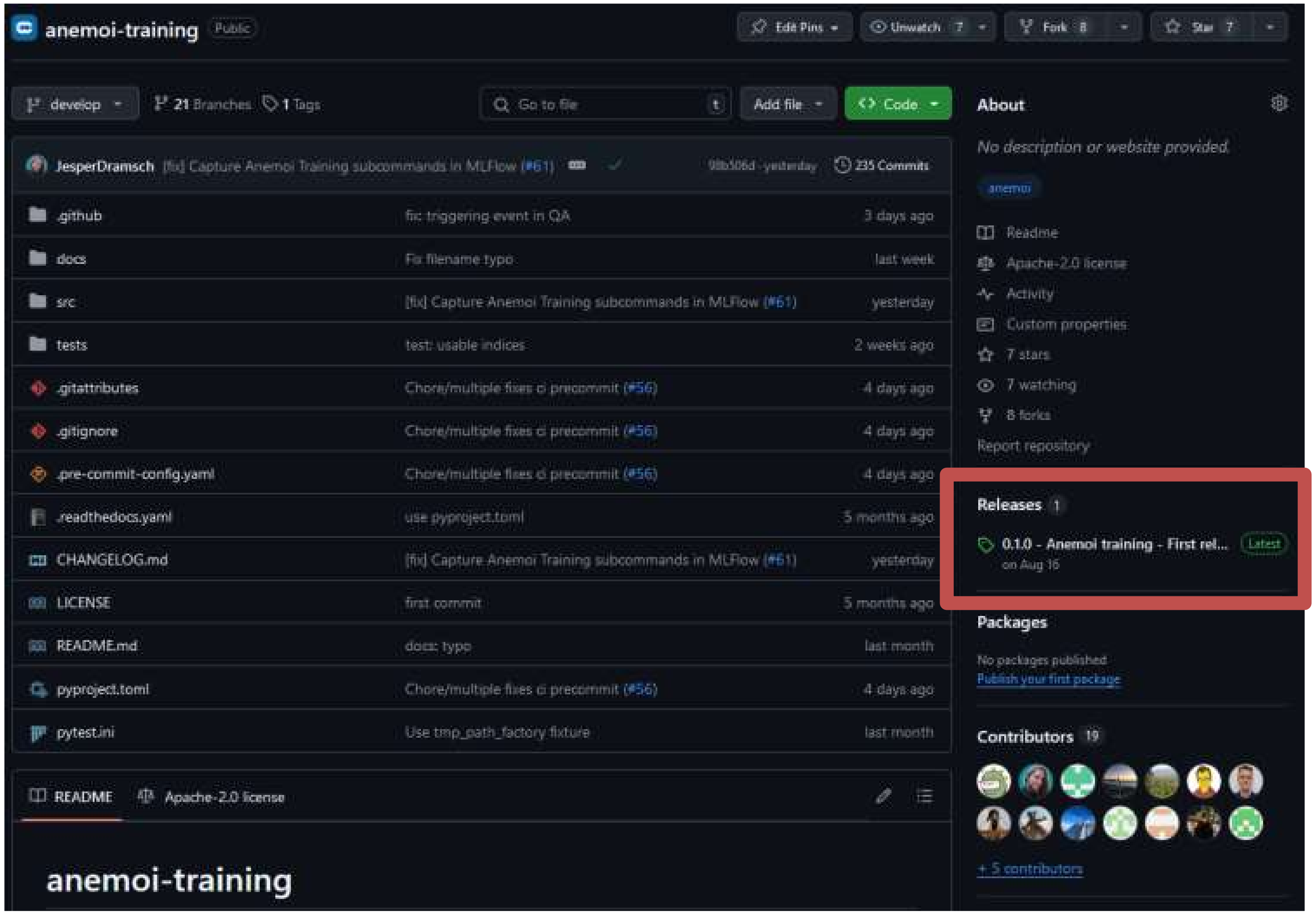
Auxiliary Work

(It's not less important; it's just non-blocking. This one is un-ordered.)

- ☑ Write the documentation on `anemoi-training`: Closed by Old: Initial Documentation for Anemoi Training ecmwf/anemoi-training#5
- ☑ Update `anemoi-models` to use `anemoi-graph`: Work in: Closed by Use anemoi-graphs HeteroData ecmwf/anemoi-models#8
- ☐ Feature parity with the `hackathon` branch `anemoi-graphs`: Tracked in https://github.com/ecmwf/anemoi-graphs/milestone/2 and https://github.com/ecmwf/anemoi-graphs/milestone/3
- ☑ Release `anemoi-models`
- ☑ Release `anemoi-graphs`
- ☑ Bring changes of attention dropout on `aifs-mono` to `anemoi-models` [illegible]

Feature: User Configs

For configs, we have the following possible features:



anemoi-training Public

JesperDramsch [fix] Capture Anemoi Training subcommands in MLFlow (#61) 98b306d · yesterday 235 Commits

.github	fic triggering event in QA	3 days ago
docs	Fix filename typo	last week
src	[fix] Capture Anemoi Training subcommands in MLFlow (#61)	yesterday
tests	test: usable indices	2 weeks ago
.gitattributes	Chore/multiple fixes ci precommit (#56)	4 days ago
.gitignore	Chore/multiple fixes ci precommit (#56)	4 days ago
.pre-commit-config.yaml	Chore/multiple fixes ci precommit (#56)	4 days ago
.readthedocs.yaml	use pyproject.toml	5 months ago
CHANGELOG.md	[fix] Capture Anemoi Training subcommands in MLFlow (#61)	yesterday
LICENSE	first commit	5 months ago
README.md	docs: typo	last month
pyproject.toml	Chore/multiple fixes ci precommit (#56)	4 days ago
pytest.ini	Use tmp_path_factory fixture	last month

anemoi-training

About

No description or website provided.

Releases 1

0.1.0 - Anemoi training - First rel... Latest
on Aug 16

Packages

No packages published
Publish your first package

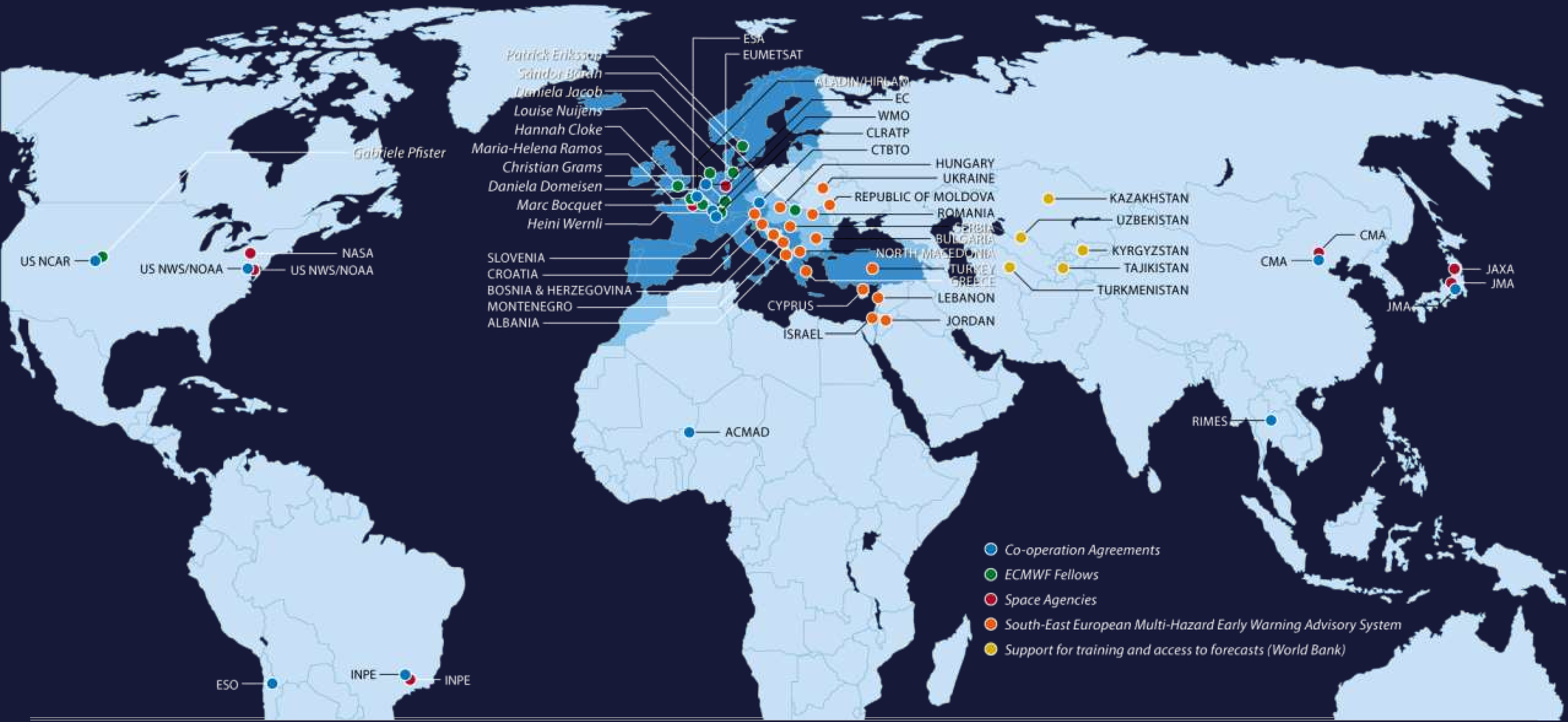
Contributors 19

+ 5 contributors

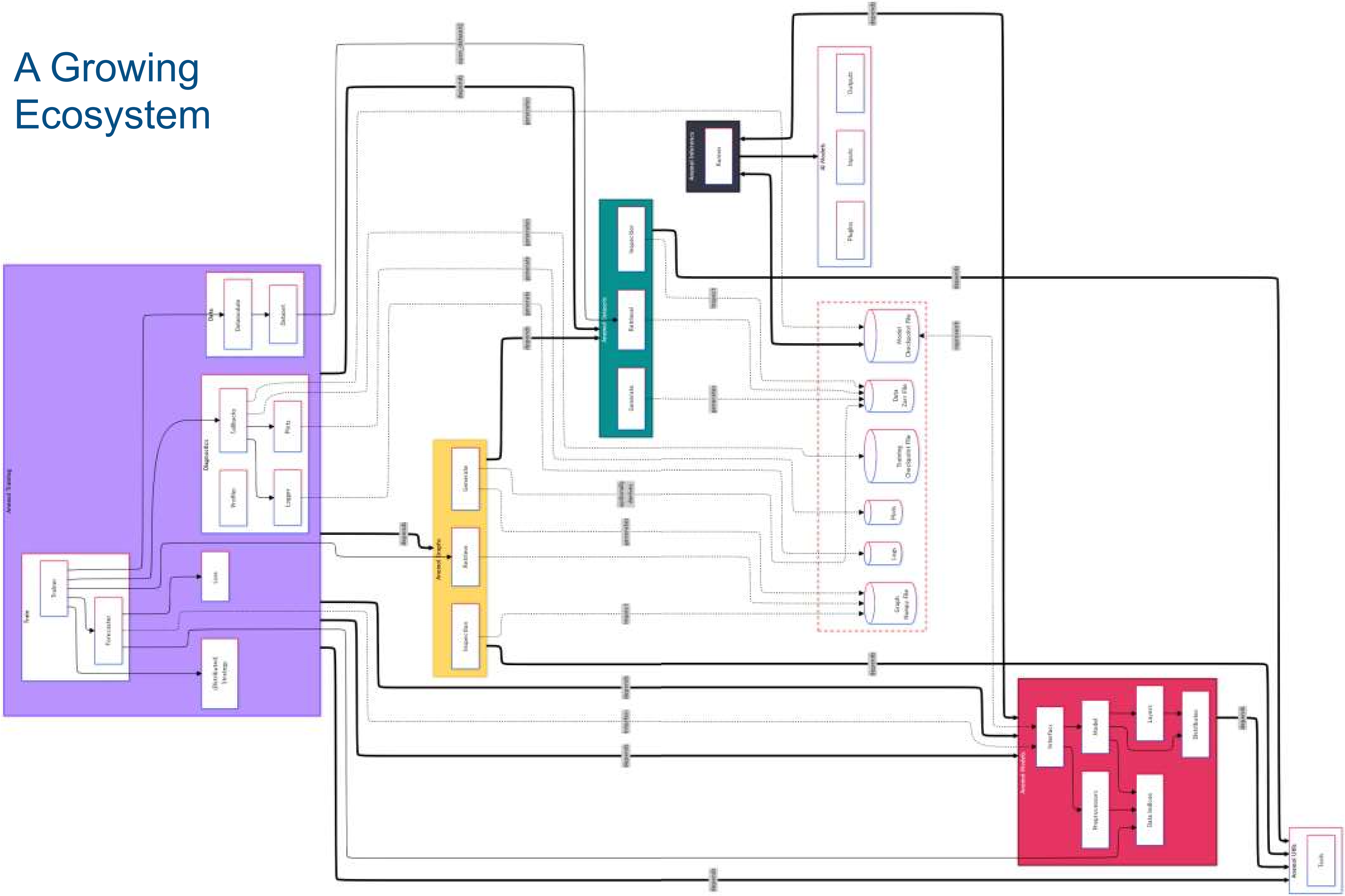
Going Global

ECMWF | EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

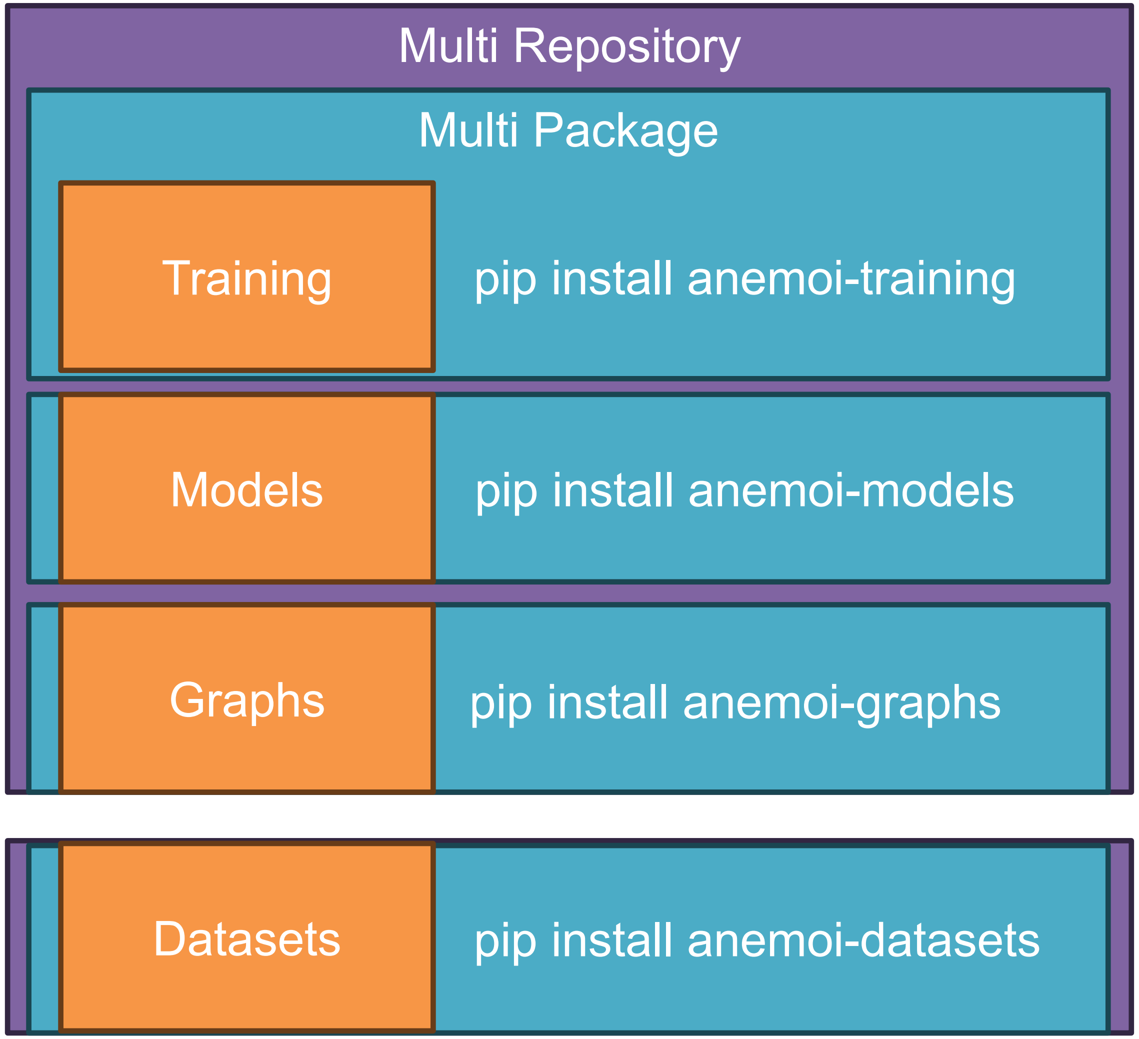
Global Collaboration



A Growing Ecosystem



Anemoi Ecosystem (“Partial” mono-repo)



Multi Repository

Multi Package

Training — pip install anemoi-training

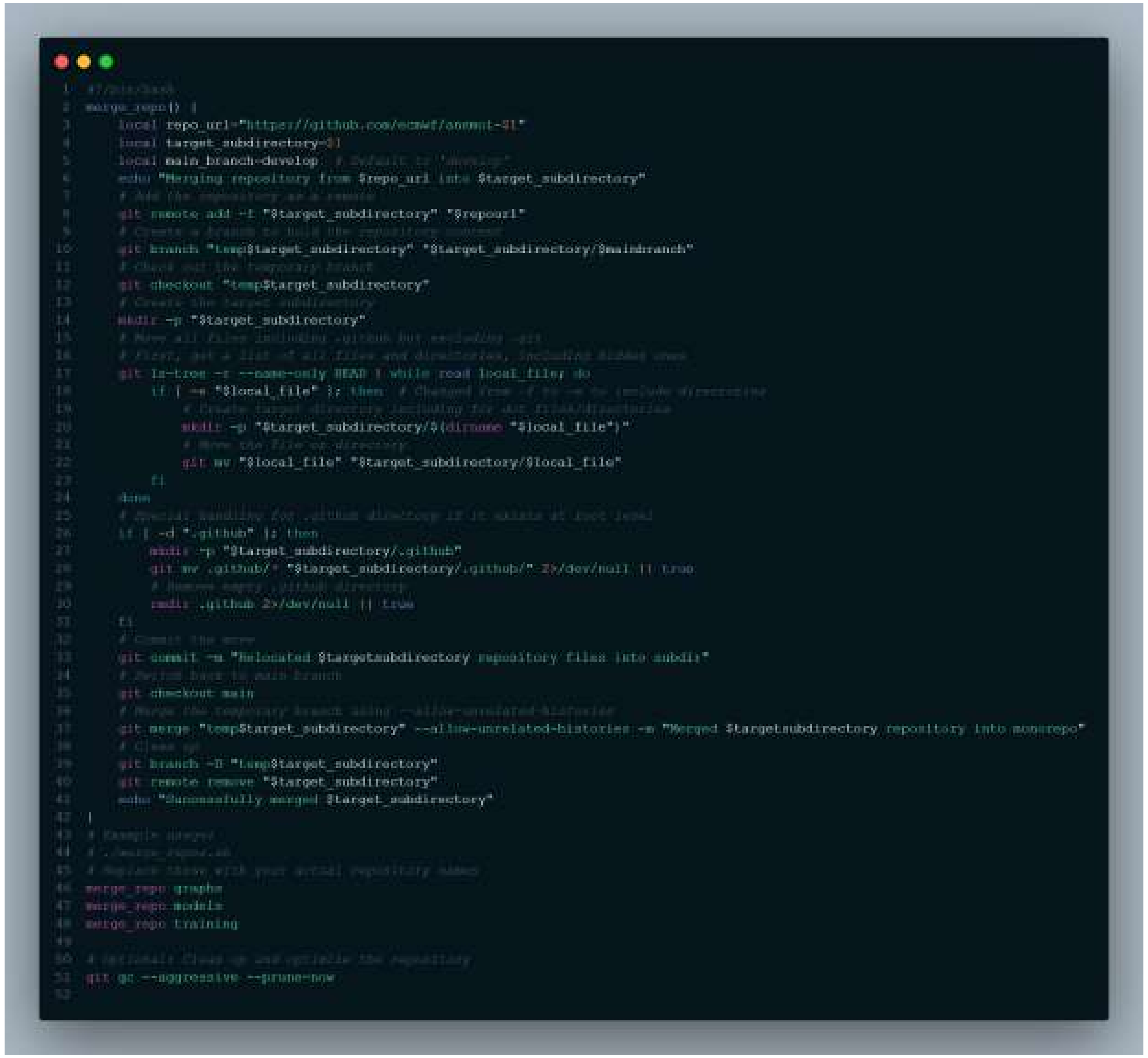
Models — pip install anemoi-models

Graphs — pip install anemoi-graphs

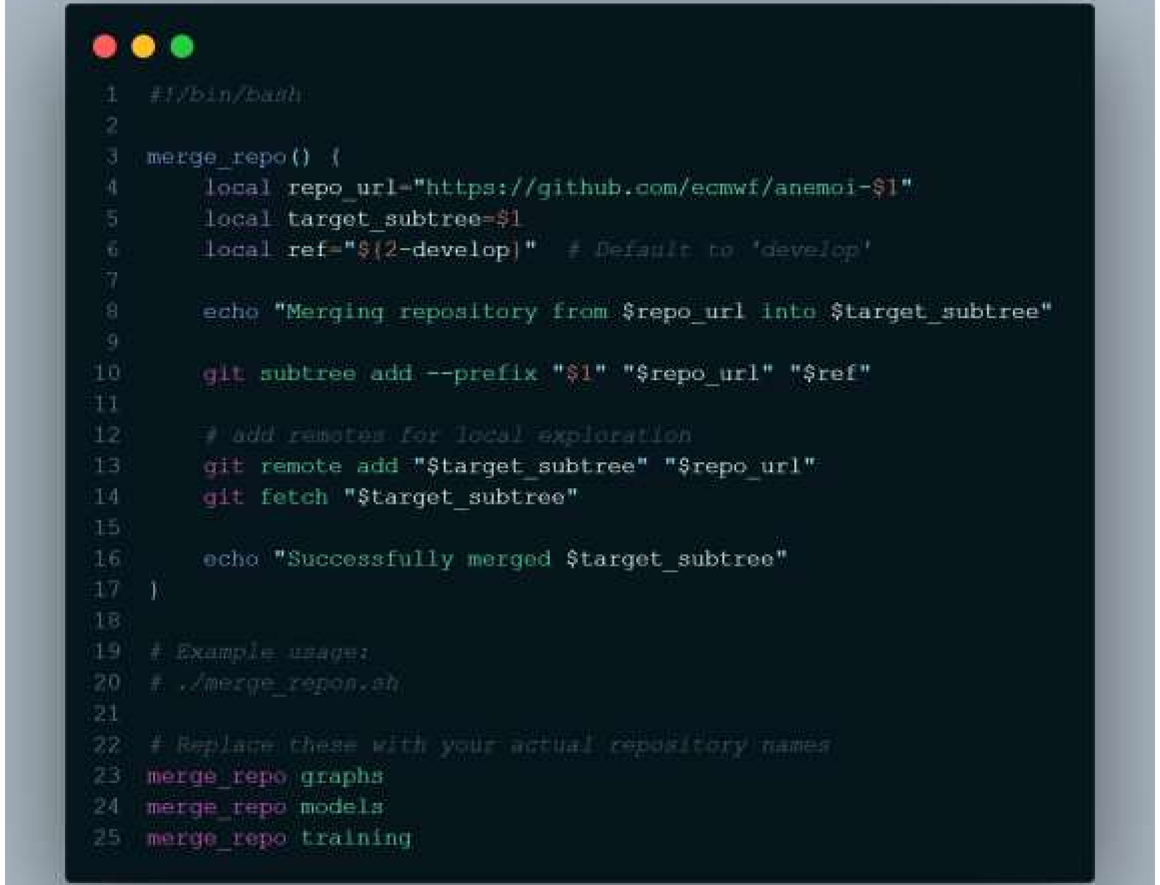
Datasets — pip install anemoi-datasets

- Pros
  - ▲ Some Separation of concerns
  - ▲ Puts tightly coupled code together
  - ▲ Different styles of collaboration possible
  - ▲ Simpler testing
  - ▲ Simpler configuration management
- Cons
  - ▼ Solves some CI but not „end-to-end“
  - ▼ Inconsistent workflows across anemoi
  - ▼ Some complex release workflows
  - ▼ Risk of creating artificial boundaries
  - ▼ Could complicate some dependencies

How AI failed me here

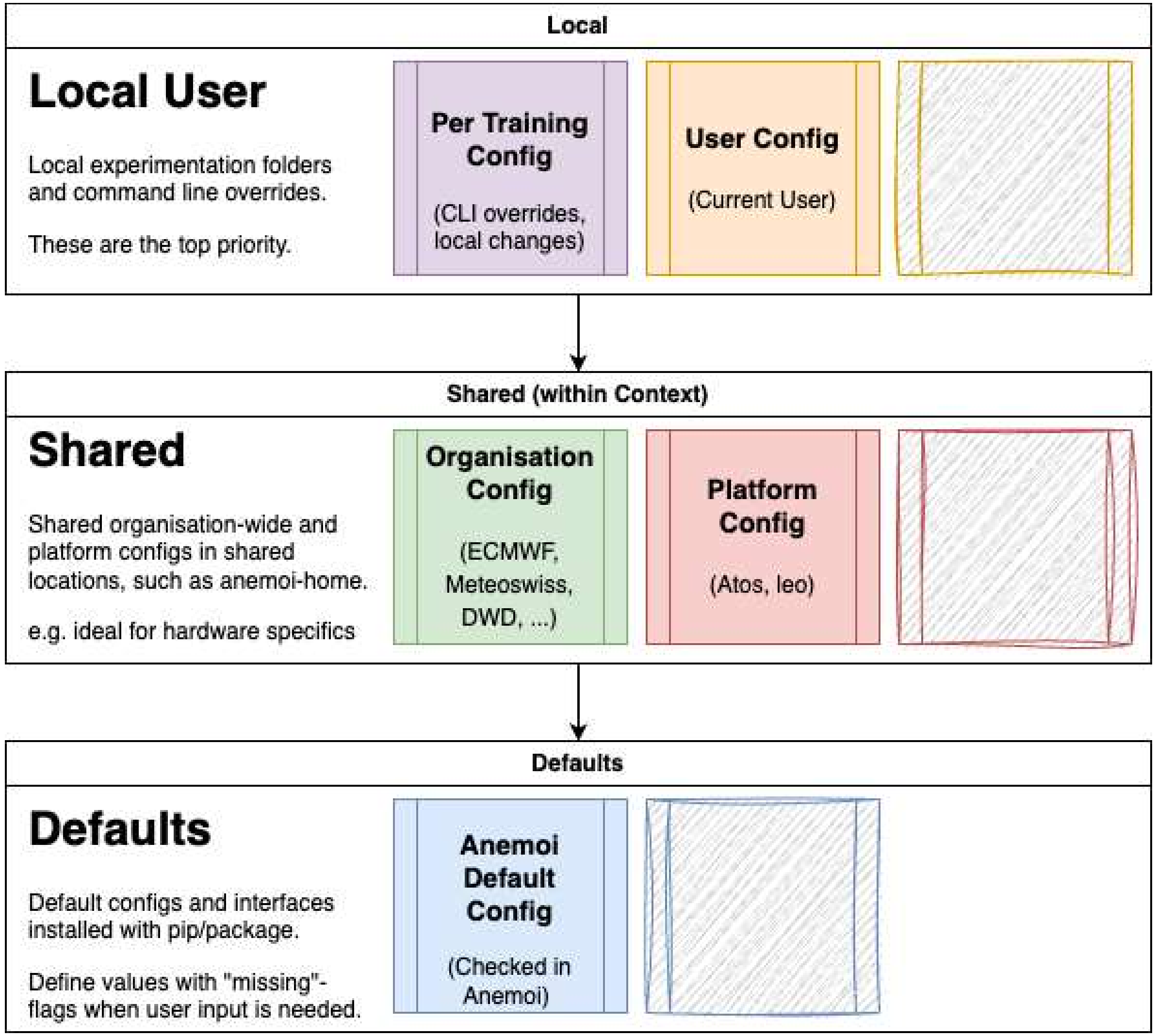


```bash
#!/bin/bash
merge_repo() {
    local repo_url="https://github.com/ecmwf/anemoi-$1"
    local target_subdirectory=$1
    local main_branch=develop  # Default to 'develop'
    echo "Merging repository from $repo_url into $target_subdirectory"
    # Add the repository as a remote
    git remote add -f "$target_subdirectory" "$repourl"
    # Create a branch to hold the repository content
    git branch "temp$target_subdirectory" "$target_subdirectory/$mainbranch"
    # Check out the temporary branch
    git checkout "temp$target_subdirectory"
    # Create the target subdirectory
    mkdir -p "$target_subdirectory"
    # Move all files including .github but excluding .git
    # First, get a list of all files and directories, including hidden ones
    git ls-tree -r --name-only HEAD | while read local_file; do
        if [ -e "$local_file" ]; then  # Changed from -f to -e to include directories
            # Create target directory including for dot files/directories
            mkdir -p "$target_subdirectory/$(dirname "$local_file")"
            # Move the file or directory
            git mv "$local_file" "$target_subdirectory/$local_file"
        fi
    done
    # Special handling for .github directory if it exists at root level
    if [ -d ".github" ]; then
        mkdir -p "$target_subdirectory/.github"
        git mv .github/* "$target_subdirectory/.github/" 2>/dev/null || true
        # Remove empty .github directory
        rmdir .github 2>/dev/null || true
    fi
    # Commit the move
    git commit -m "Relocated $targetsubdirectory repository files into subdir"
    # Switch back to main branch
    git checkout main
    # Merge the temporary branch using --allow-unrelated-histories
    git merge "temp$target_subdirectory" --allow-unrelated-histories -m "Merged $targetsubdirectory repository into monorepo"
    # Clean up
    git branch -D "temp$target_subdirectory"
    git remote remove "$target_subdirectory"
    echo "Successfully merged $target_subdirectory"
}
# Example usage:
# ./merge_repos.sh
# Replace these with your actual repository names
merge_repo graphs
merge_repo models
merge_repo training

# Optional: Clean up and optimize the repository
git gc --aggressive --prune=now

```



```bash
#!/bin/bash

merge_repo() {
    local repo_url="https://github.com/ecmwf/anemoi-$1"
    local target_subtree=$1
    local ref="${2-develop}"  # Default to 'develop'

    echo "Merging repository from $repo_url into $target_subtree"

    git subtree add --prefix "$1" "$repo_url" "$ref"

    # add remotes for local exploration
    git remote add "$target_subtree" "$repo_url"
    git fetch "$target_subtree"

    echo "Successfully merged $target_subtree"
}

# Example usage:
# ./merge_repos.sh

# Replace these with your actual repository names
merge_repo graphs
merge_repo models
merge_repo training
```

Extend Configs to Institutional levels



Local

Local User

Local experimentation folders and command line overrides.

These are the top priority.

Per Training Config

(CLI overrides, local changes)

User Config

(Current User)

Shared (within Context)

Shared

Shared organisation-wide and platform configs in shared locations, such as anemoi-home.

e.g. ideal for hardware specifics

Organisation Config

(ECMWF, Meteoswiss, DWD, ...)

Platform Config

(Atos, leo)

Defaults

Defaults

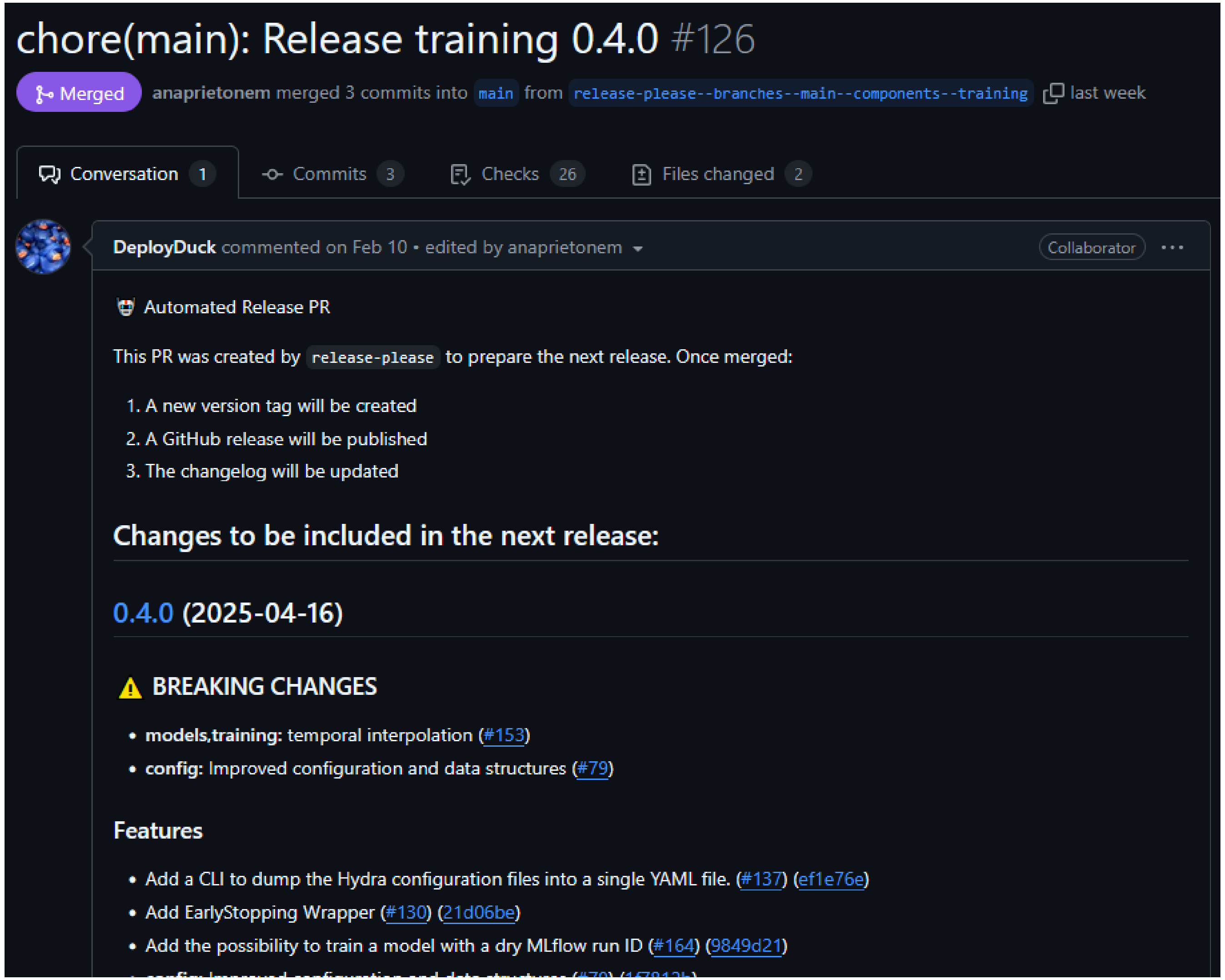
Default configs and interfaces installed with pip/package.

Define values with "missing"-flags when user input is needed.

Anemoi Default Config

(Checked in Anemoi)

Automate the Release process



chore(main): Release training 0.4.0 #126

anaprietonem merged 3 commits into `main` from `release-please--branches--main--components--training` last week

DeployDuck commented on Feb 10 • edited by anaprietonem

Automated Release PR

This PR was created by `release-please` to prepare the next release. Once merged:

1. A new version tag will be created
2. A GitHub release will be published
3. The changelog will be updated

Changes to be included in the next release:

0.4.0 (2025-04-16)

⚠ BREAKING CHANGES

- **models,training:** temporal interpolation (#153)
- **config:** Improved configuration and data structures (#79)

Features

- Add a CLI to dump the Hydra configuration files into a single YAML file. (#137) (ef1e76e)
- Add EarlyStopping Wrapper (#130) (21d06be)
- Add the possibility to train a model with a dry MLflow run ID (#164) (9849d21)
- config: Improved configuration and data structures (#79) (1f7812b)

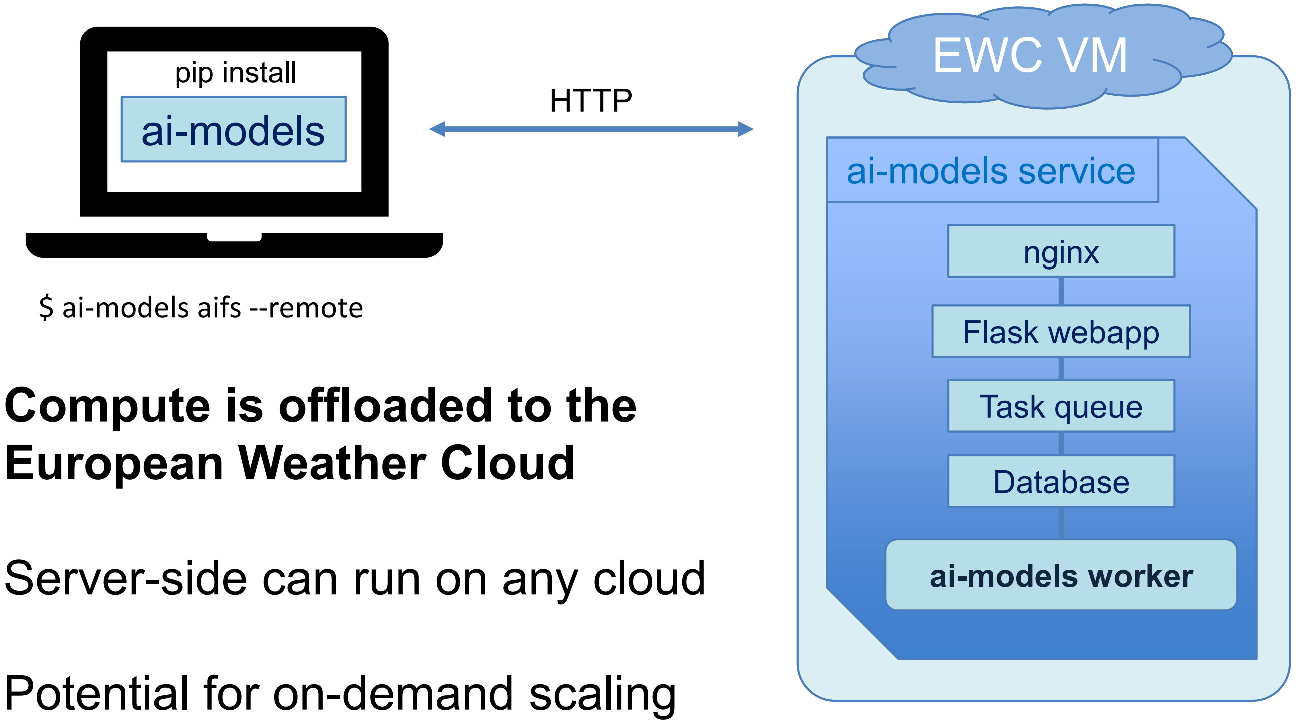
External and Global Collaboration

- Kick things off with a ”hackathon”
  - You can even do a Code Tour at a hackathon!
- Maintain roadmaps / Kanban boards
- Code Reviews of PRs are essential
  - Consider dual reviews:
    - For code quality
    - For scientific validity / business case
- Maintain configurability, modularity and extensibility
- Automate what you can to keep morale high
- Don’t be afraid to jump on calls

Success stories!

ECMWF | EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

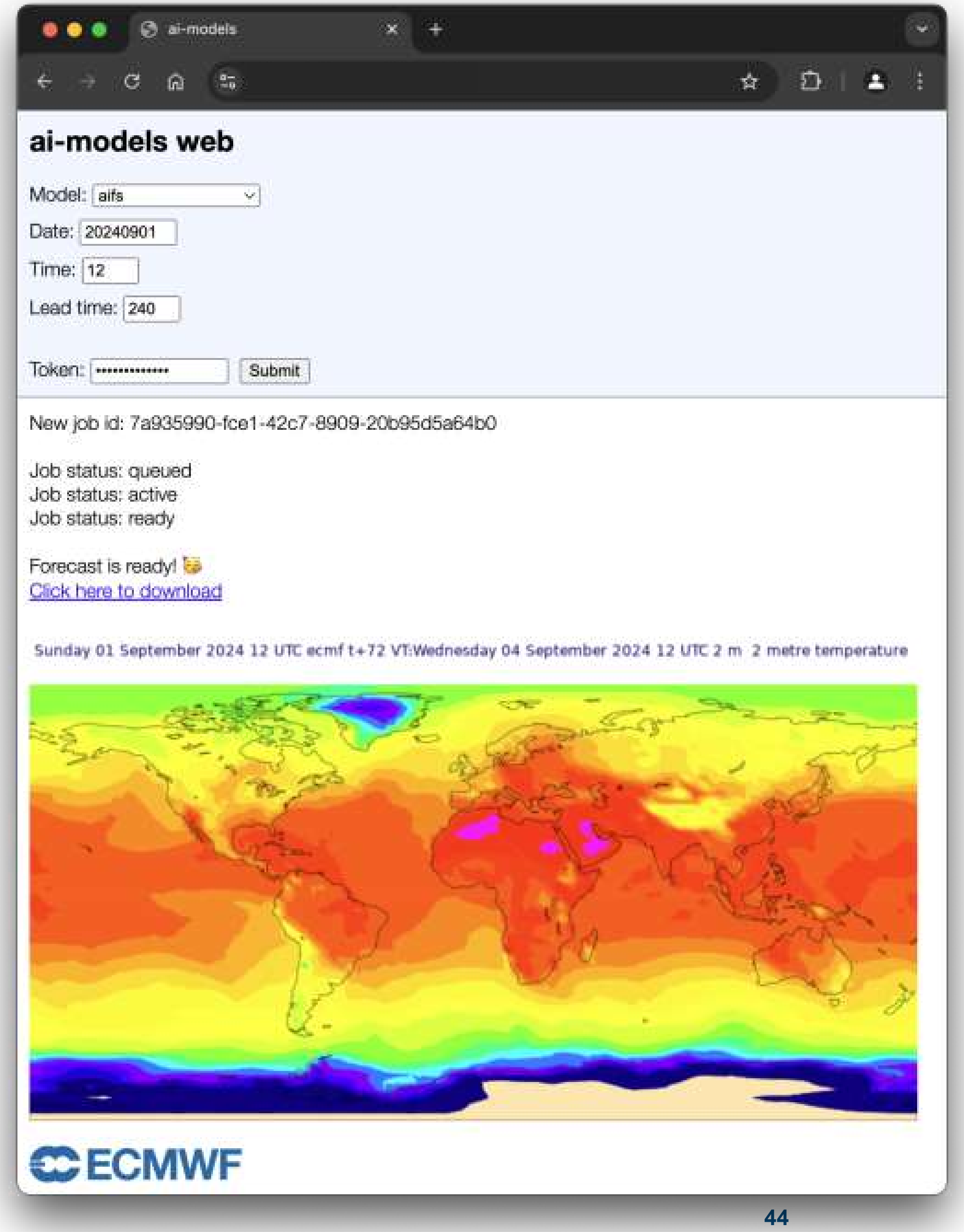
Running AIFS anywhere



```
$ ai-models aifs --remote
```

- **Compute is offloaded to the European Weather Cloud**
- Server-side can run on any cloud
- Potential for on-demand scaling
- Forecast-in-a-box PoC

Provided continuity for AIFS and the other ML models during 2 GPU maintenance windows this year



ai-models web

Model: aifs

Date: 20240901

Time: 12

Lead time: 240

Token: •••••••••••••• Submit

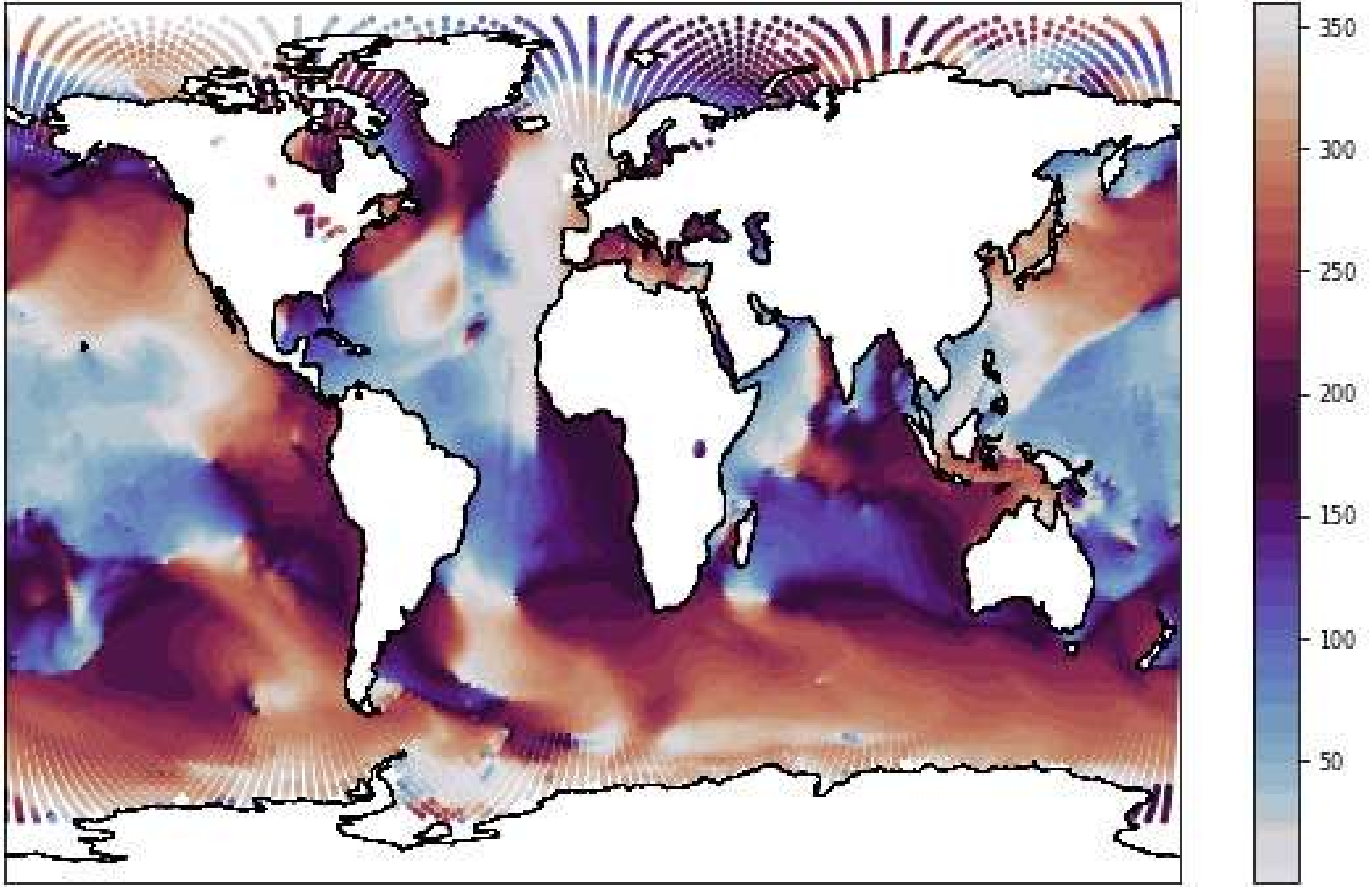
New job id: 7a935990-fce1-42c7-8909-20b95d5a64b0

Job status: queued
Job status: active
Job status: ready

Forecast is ready! 🥳
Click here to download

Sunday 01 September 2024 12 UTC ecmf t+72 VT:Wednesday 04 September 2024 12 UTC 2 m 2 metre temperature

Adding in Earth System Components

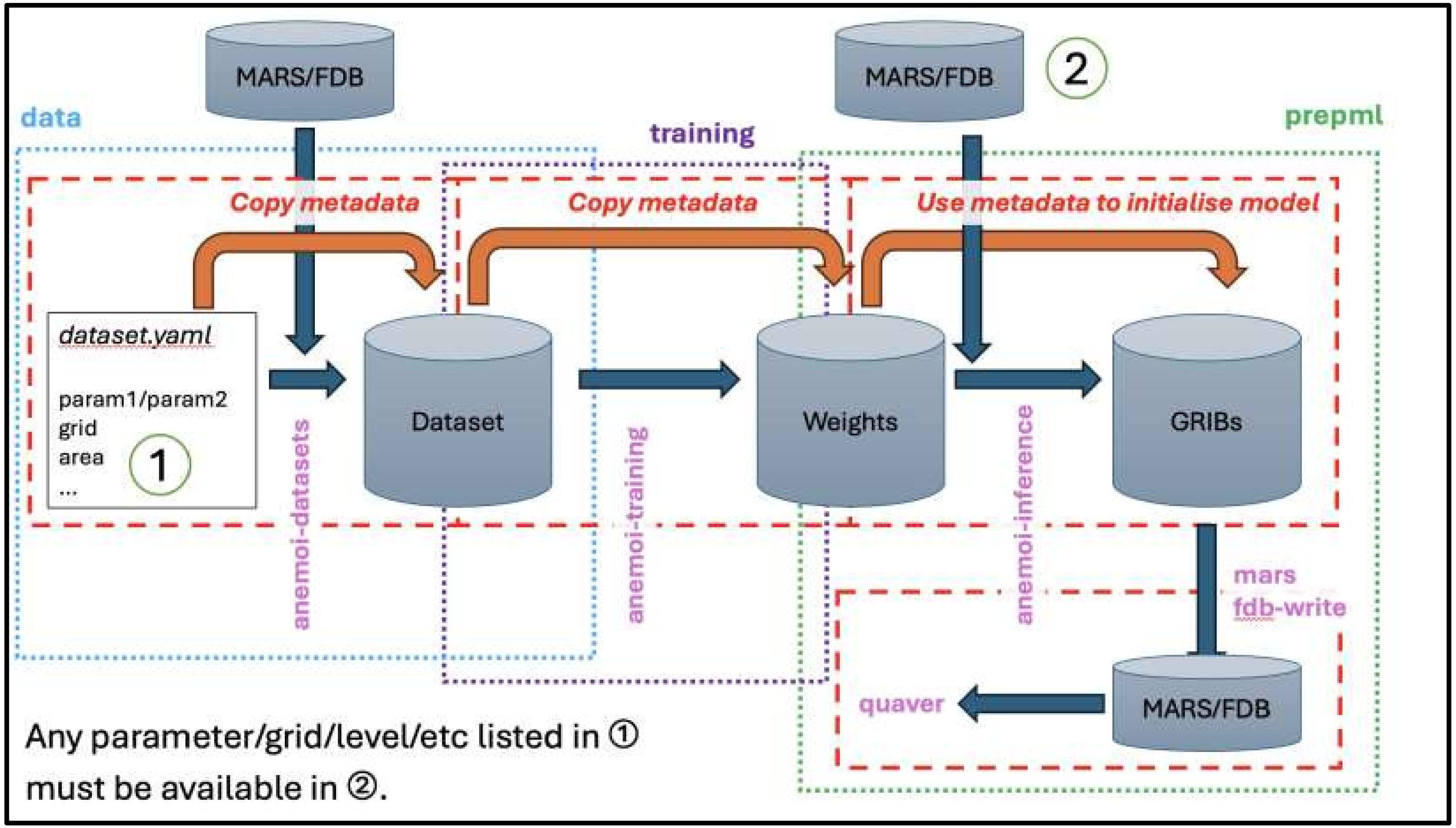


the lower the better

Mean wave direction (14 day forecast)

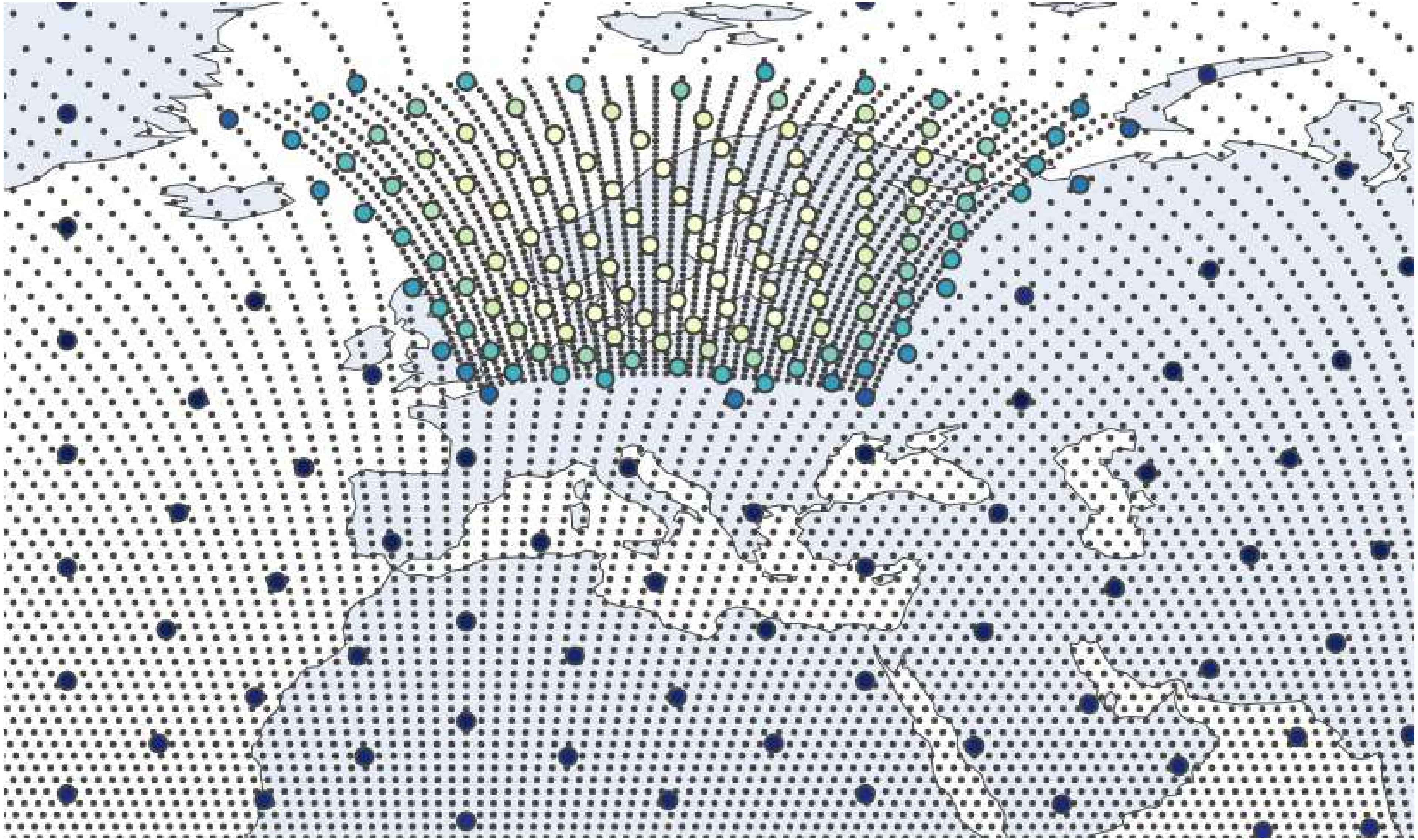
Tracking metadata through the entire ecosystem

All this is possible thanks to tracking the metadata.

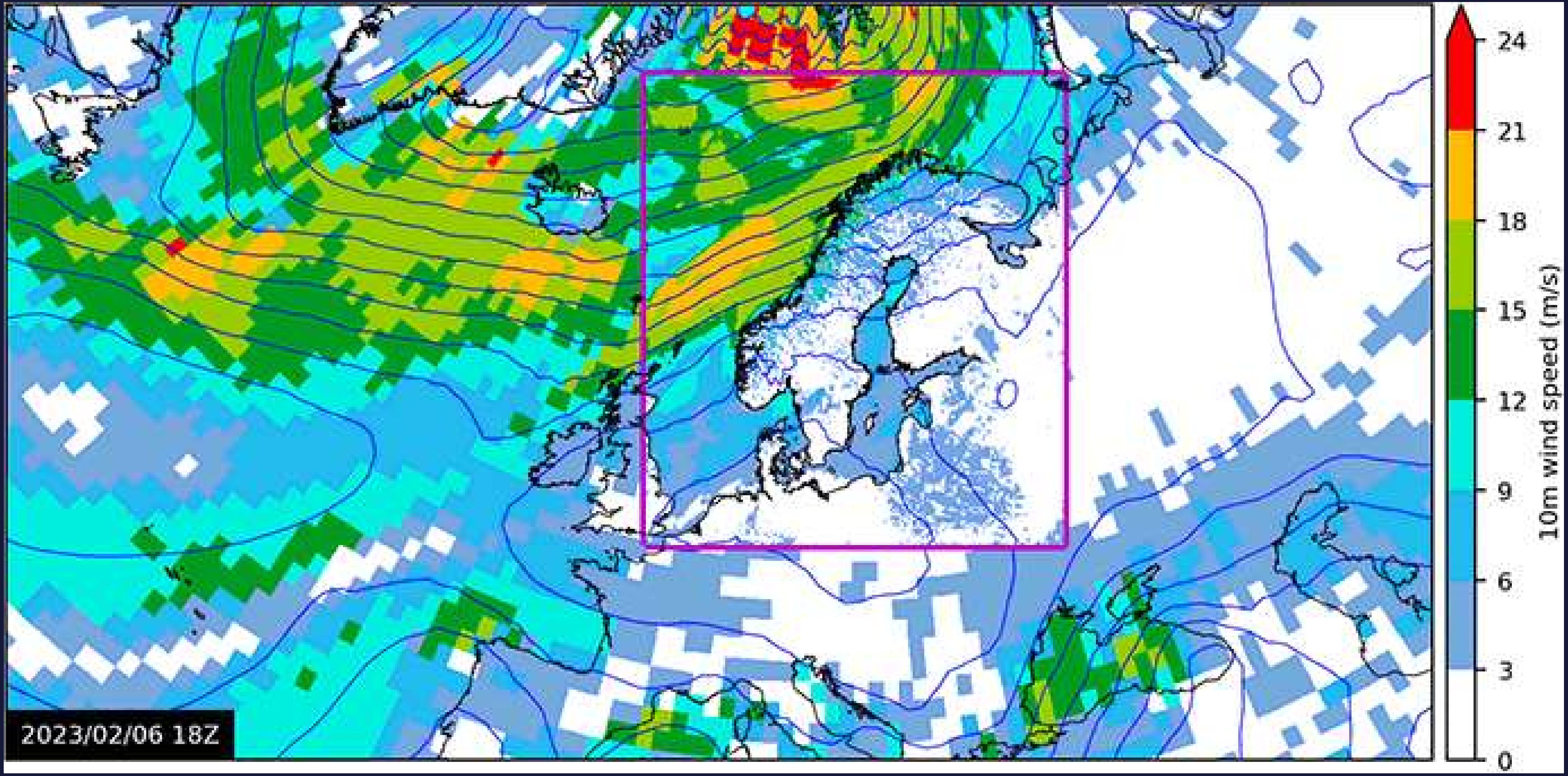


Any parameter/grid/level/etc listed in ①
must be available in ②.

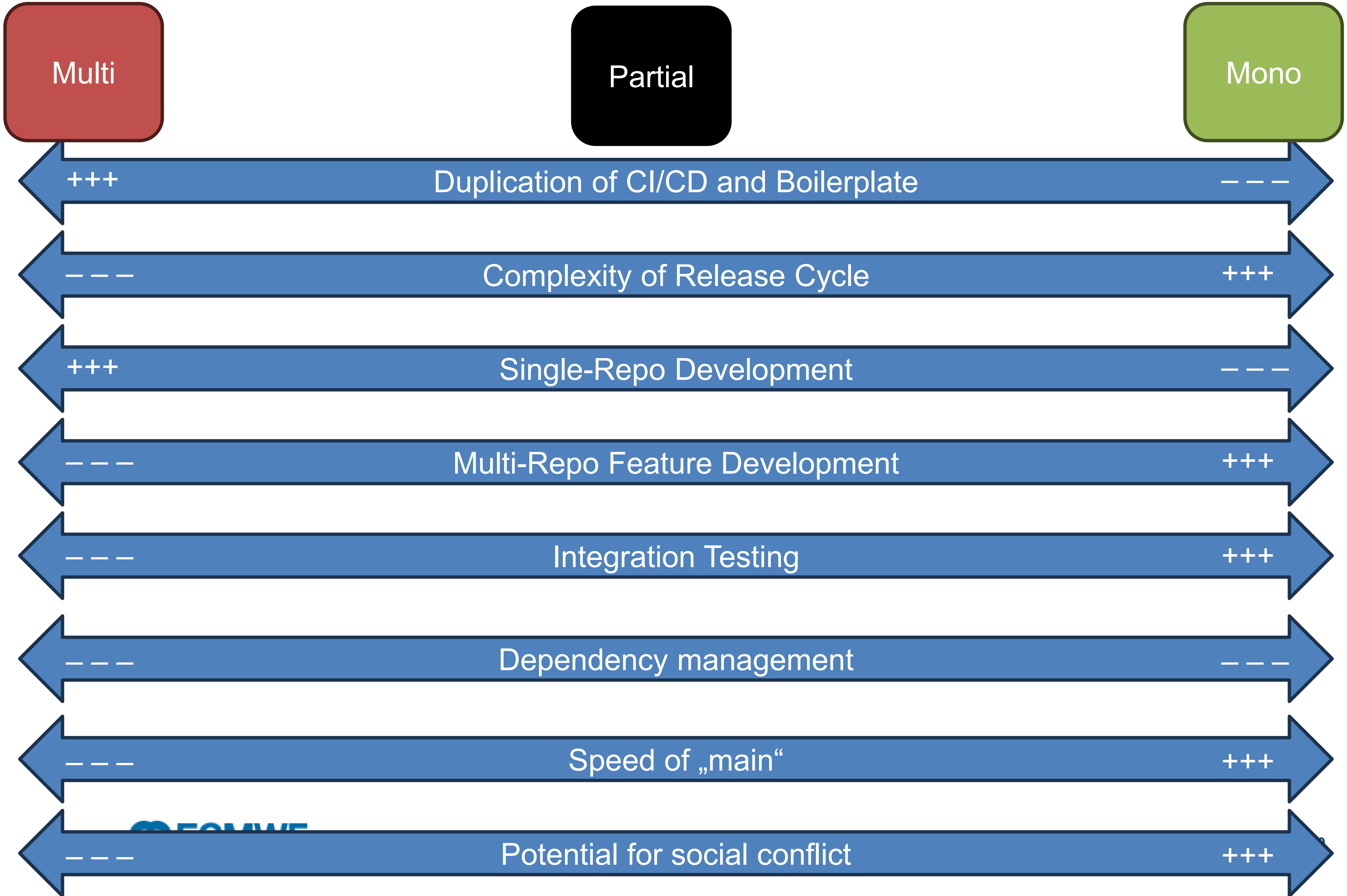
Enabling many types of graphs



Enabling many types of graphs (courtesy Met Norway)



Takeaways



	Multi	Partial	Mono
Duplication of CI/CD and Boilerplate	+++		– – –
Complexity of Release Cycle	– – –		+++
Single-Repo Development	+++		– – –
Multi-Repo Feature Development	– – –		+++
Integration Testing	– – –		+++
Dependency management	– – –		– – –
Speed of „main“	– – –		+++
Potential for social conflict	– – –		+++

Learnings for growing software projects

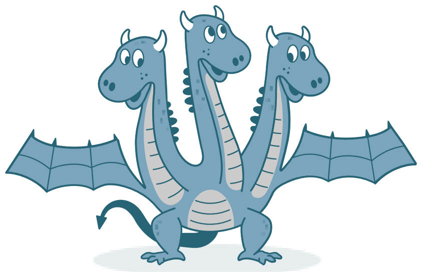
- Get yourself a software architect or two
- Do Code Tours and in-person events
- Anticipate user needs and extensibility requirements
- Make decisions that are easy to reverse
- Make design choices that are easy to extend
- Don’t just let AI agents run through your code
- Repo structures are difficult; use tools to make it easier
- Keep dependencies light, they are hell after all
- Collaborate and listen!

Core Tech Stack

NumPy

Zarr

PyG



PyTorch

mlflow™

PyTorch Lightning

einops

Pydantic

Bringing schema and sanity to your data