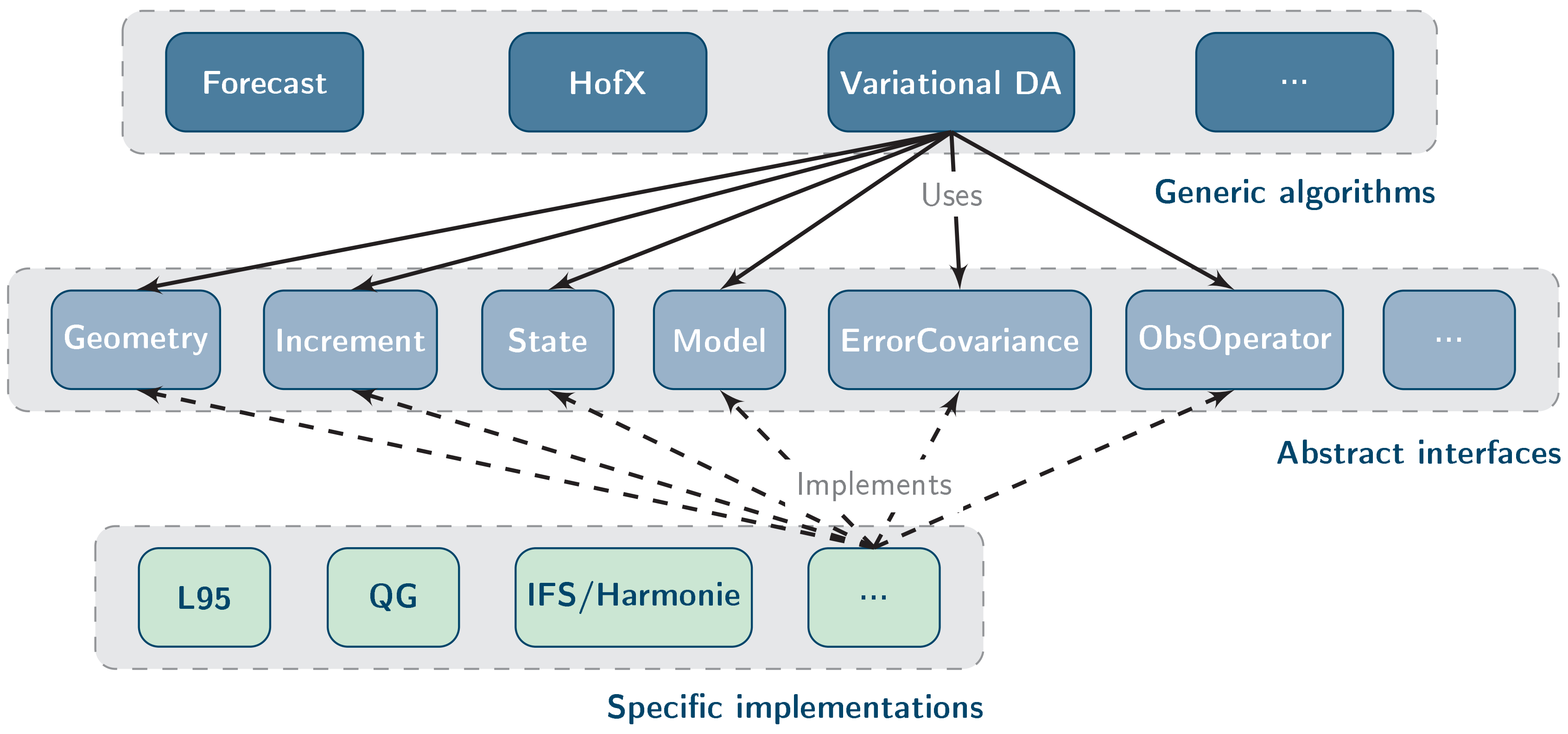
Making OOPS easier to use for new systems

Benjamin Ménétrier - Norwegian Meteorological Institute

QUENCH, a new generic model in the OOPS framework

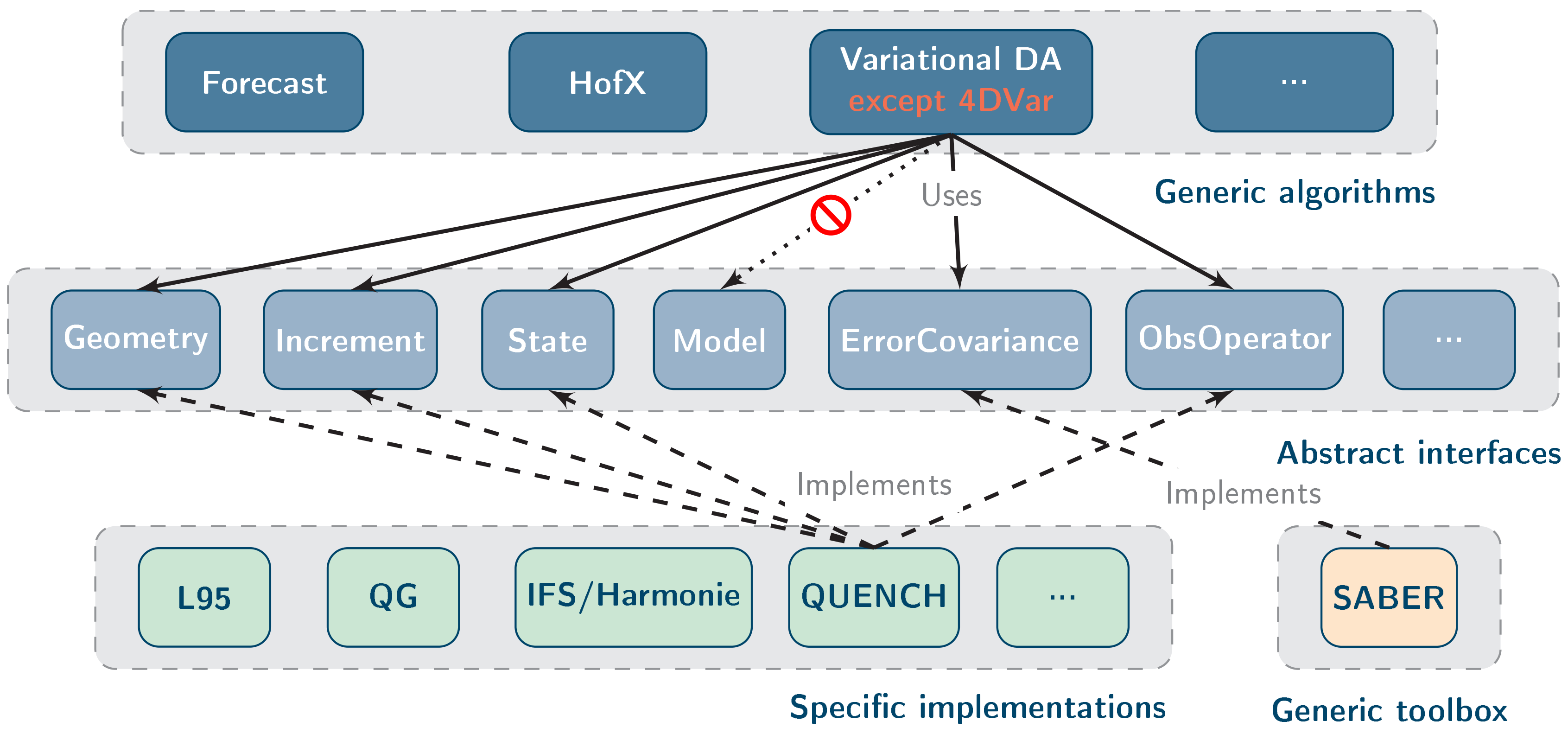
OOPS design is based on the separation of concerns:

- The generic algorithms use abstract interfaces.
- Specific implementations of these interfaces are needed for each model.
- Main advantage: any new feature added at the generic algorithms level is immediately available for all the models.
- Main drawback: writing new specific implementations to interface an existing system with OOPS can take a lot of time, up to a decade (IFS/Harmonie).



QUENCH is a new model in the OOPS framework:

- Generic geometry and fields based on ATLAS.
- Generic background error covariance with SABER.
- No time propagation: 3D-Var, 3D-EnVar and 4D-EnVar only.
- Main advantage: QUENCH is generic and can be used for many systems and various grids (e.g. surface T/q, land surface, ocean).
- Main drawback: 4DVar is not available.



QUENCH characteristics

- Written in C++ (∼ 5,700 lines), MPI parallelization.
- Based on ATLAS for geometry and fields (easy interface with Fortran).
- Can handle masks, vertically staggered grids, orography, etc.
- Already available by default:
  - NetCDF I/O for background and observations files.
  - Bilinear interpolation (+ adjoint) for the observation operator.

Newcomer TODO list

You would like to benefit from the OOPS framework with your own system?

- Optionally add your own I/O class to read background and observations files.
- Optionally implement your own observation operator (+TL/AD).
- Create a background error covariance matrix using the multiple options of the SABER library (`https://github.com/JCSDA/saber`).
- Adapt the json configuration file for your geometry, your files paths, etc.

Installation requirements

QUENCH can run on laptops for small applications and on HPC systems for larger ones. To install it, you need:

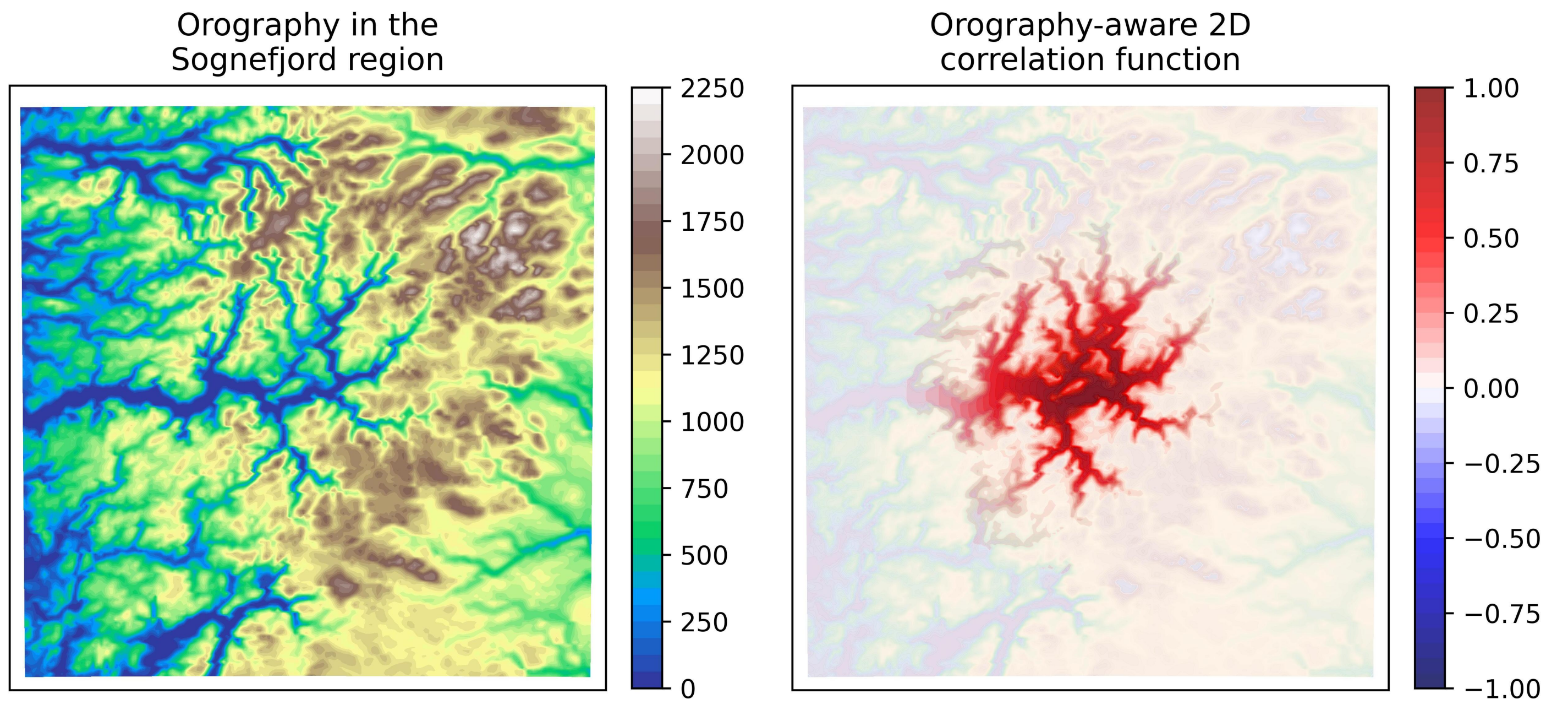
- The latest version of OOPS (access should be requested on ECMWF website):
  → `https://git.ecmwf.int/projects/OOPS/repos/oops`
- ECSABER, the interface of SABER with the ECMWF version of OOPS:
  → `https://git.ecmwf.int/projects/SAB/repos/ecsaber`

Support can be provided: benjamin.menetrier@met.no

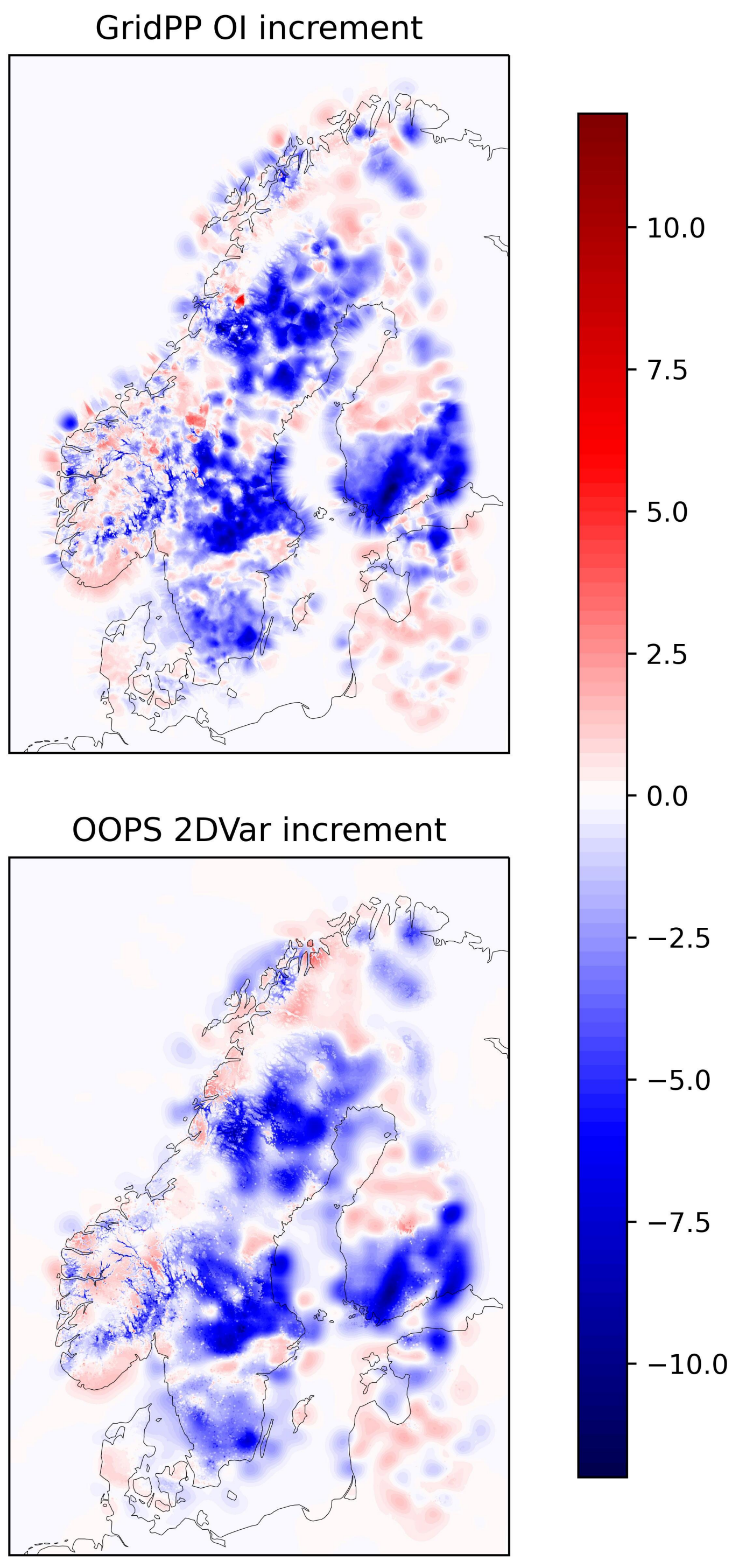
Test case: high-resolution surface temperature analysis

Preliminary comparison of the GridPP OI and OOPS 2DVar:

- Background: 2m temperature MEPS forecast downscaled at 1km resolution with GridPP.
- Observations: Netatmo surface network, 50 times denser than the usual networks of national weather services.
- 2DVar background error covariance based on SABER:
  - Locally varying standard-deviation.
  - Orography-aware inhomogeneous 2D correlation.
- Local diagnostic of standard-deviation and horizontal length-scale based on the Hollingsworth-Lönnberg method.
- Similar analysis RMSE with respect to control observations.



Dirac test with the orography-aware 2D correlation function



Temperature increments (K)

Conclusions

- The QUENCH model enables a purely file-based use of OOPS for existing systems, without requiring a time-consuming development of interfaces.
- It can achieve good performances using the SABER library for background error covariance.

Future work

- Testing 2DEnVar for surface T/q.
- Testing new systems like land surface (SURFEX).
- Including more complex observations.
- Exploring coupled DA, by combining IFS/Harmonie and QUENCH at the OOPS interface level.

Model vs. system?

In this poster, the term *model* refers to the OOPS template parameter (e.g. L95, QG, IFS or QUENCH). It does not refer to the *Model* abstract class of OOPS, which deals with the time propagation of fields. Conversely, the term *system* refers to any forecasting software in a general sense.