Hands-on session “users”

Introduction

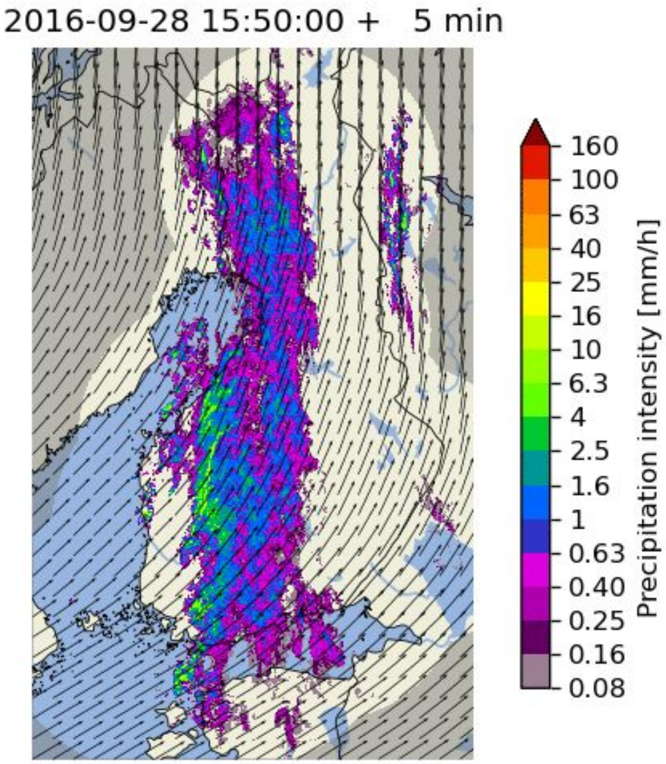
Overview of pysteps modules

- module flowchart with short description
- .
- .

Nowcast workflow

Optical Flow and Extrapolation

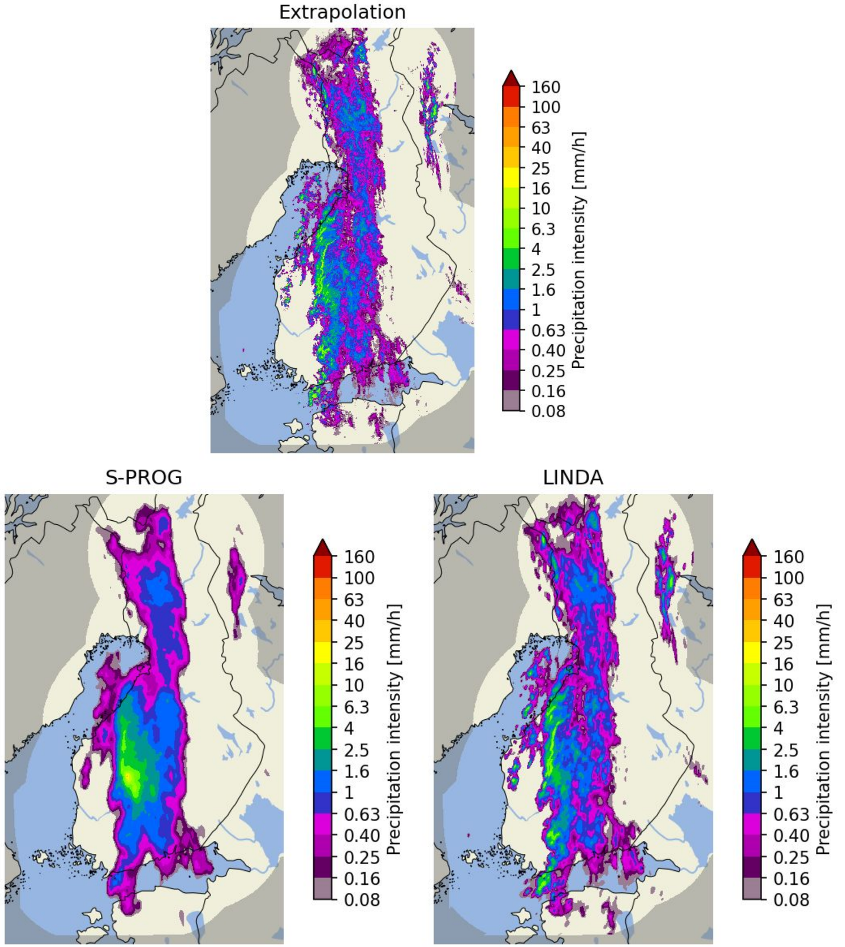
- Advection (optical flow & extrapolation) is the key component of all pysteps nowcasting methods.
- All methods are based on the “Lagrangian persistence” nowcast shown on the right
- Three different types of optical flow methods have been implemented in the **motion** module:
    - Feature tracking: Lucas-Kanade
    - Variational: VET and Proesmans
    - Spectral: DARTS
- For advection, pysteps implements the backward semi-Lagrangian scheme in the **extrapolation** module.



Deterministic Nowcasts

The main methods implemented in the **nowcasts** module:

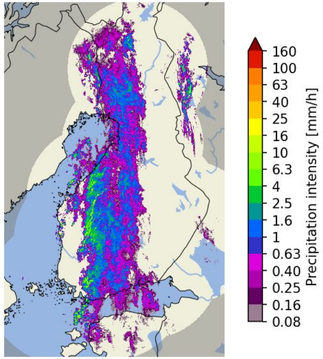
Method	**Pros**	**Cons**	**Typical computation time**
Extrapolation	very fast	no prediction of growth or decay of precipitation	< 10 seconds
S-PROG	● for low-intensity precipitation (< 1-2 mm/h) has generally the best skill ● choose for stratiform events	inability to preserve the spatial structure of rainfall fields, and particularly convective cells	< 20 seconds
LINDA-D	● the most accurate method for intense precipitation (> 1-2 mm/h) ● choose for convective events	slow to compute	might take several minutes



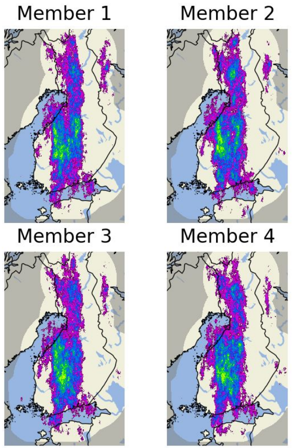
Ensemble Nowcasts

- The main ensemble methods implemented in the **nowcasts** module are STEPS and LINDA-P.
- They model two sources of uncertainty: advection field estimation and Lagrangian growth and decay
- The basic rule for choosing the method:
  - stratiform events: STEPS
  - convective events: LINDA-P
- LINDA-P generally produces more realistic ensemble members, but this comes at a high computational cost
- Computation times for the 4-member ensembles shown on the right:
  - STEPS: ~20 seconds
  - LINDA-P: ~5 minutes

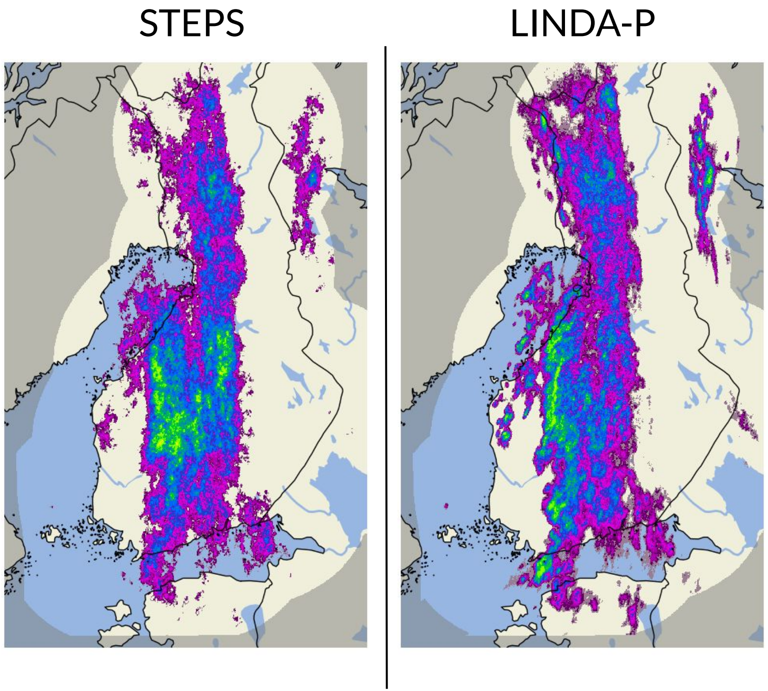
Observations at
2016-09-28 15:50 UTC



Nowcast ensemble

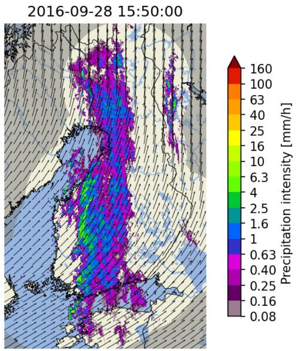


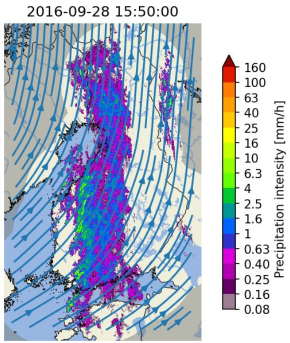
First ensemble members



Visualization tools

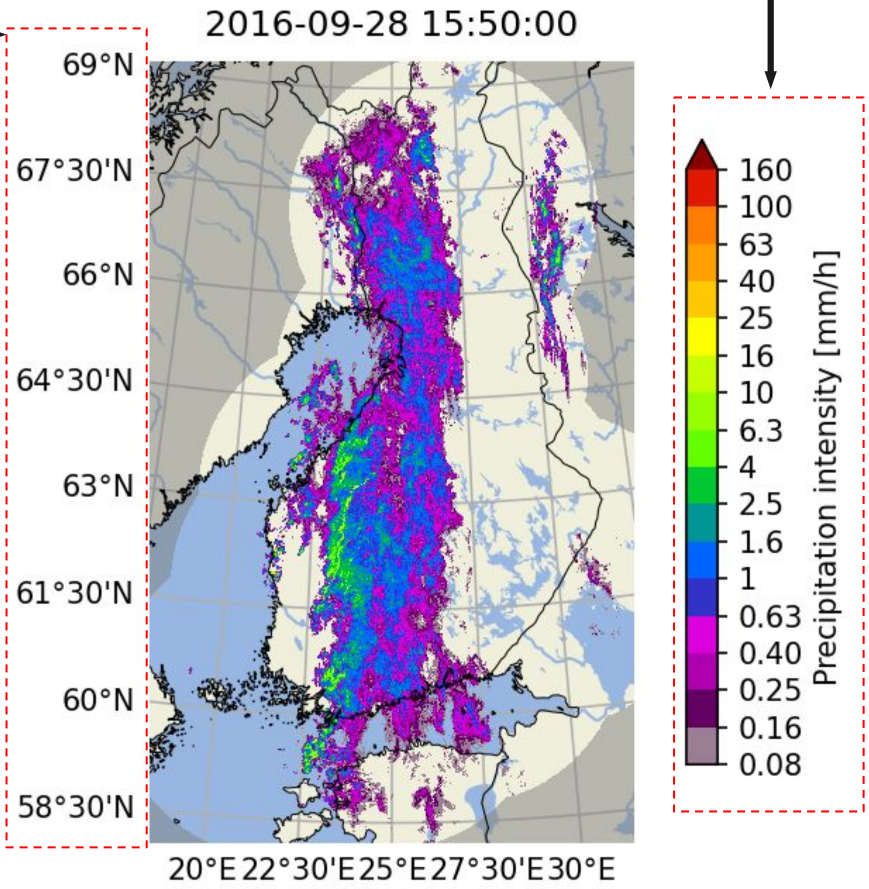
- Extensive set of visualization tools has been implemented in the **visualization** module
- Support for multiple layers: basemap, precipitation and motion field
  - Plotting of basemaps by using cartopy
  - Quivers and streamlines for advection fields





Colorbars with several pre-configured scales and for different data units

Longitude-latitude lines with labels



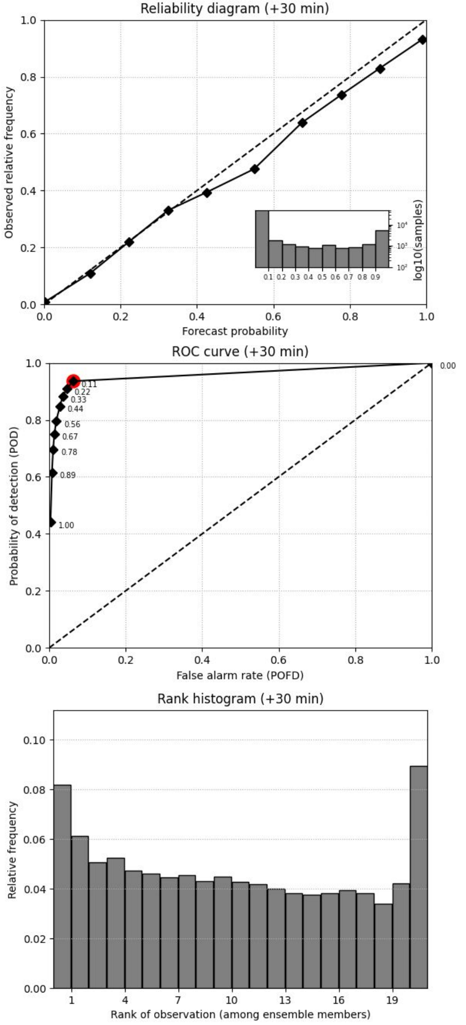
Verification tools / metrics

- A large number of verification utilities and metrics have been implemented in the **verification** module
- Functionality
    - Creation of verification objects and aggregation from multiple nowcasts
    - Plotting of verification results

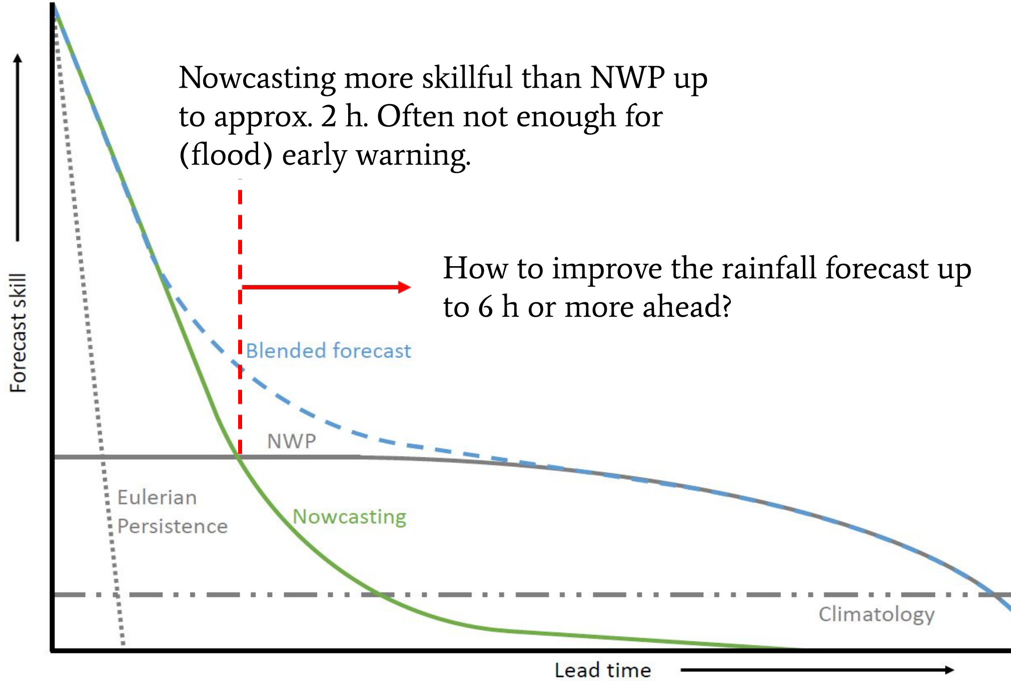
Metrics

- Deterministic
    - Categorical: CSI, ETS, POD, FAR
    - Continuous: MAE, ME
    - Scale/intensity-based metrics: FSS, intensity-scale
    - Radially averaged power spectral density (RAPSD)
- Probabilistic
    - CRPS
    - Reliability diagram
- Ensemble
    - Spread
    - Rank histogram

Examples of verification plots for 30-minute STEPS nowcasts

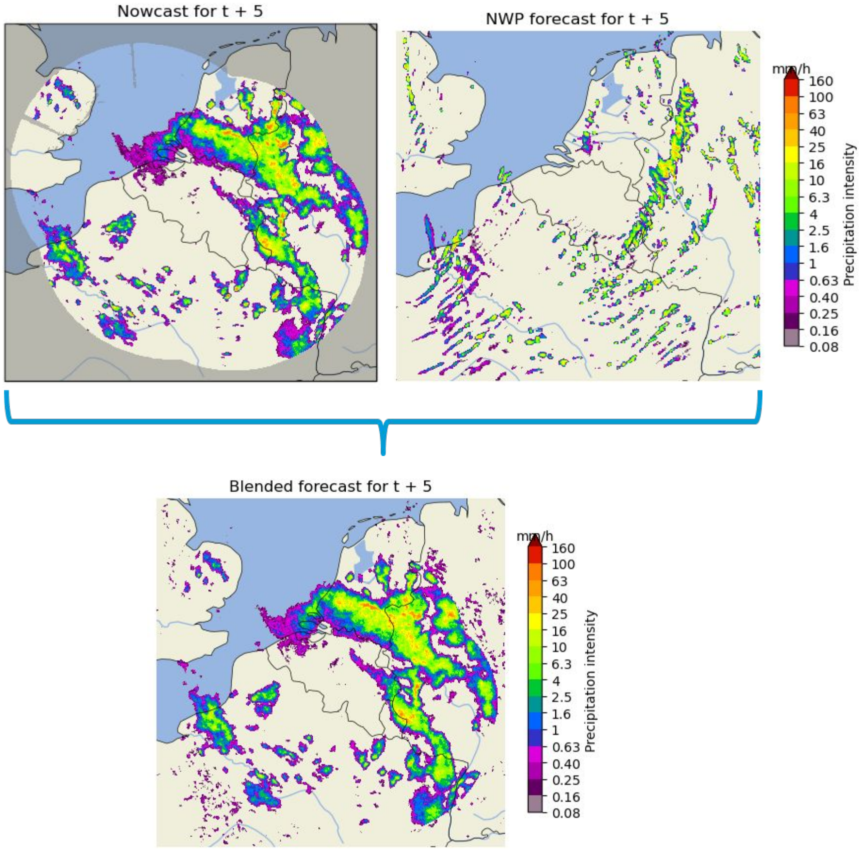


Blending with NWP



Nowcasting more skillful than NWP up to approx. 2 h. Often not enough for (flood) early warning.

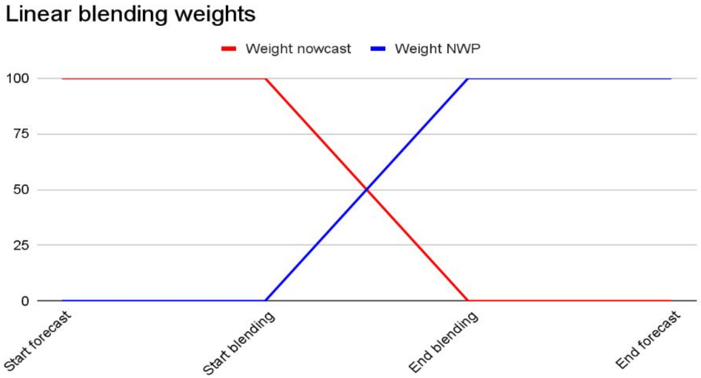
How to improve the rainfall forecast up to 6 h or more ahead?



Blending with NWP: methods in pysteps

1. Linear blending

- Fixed start and end point of blending procedure
- Weights go linearly from 1 to 0 and 0 to 1.

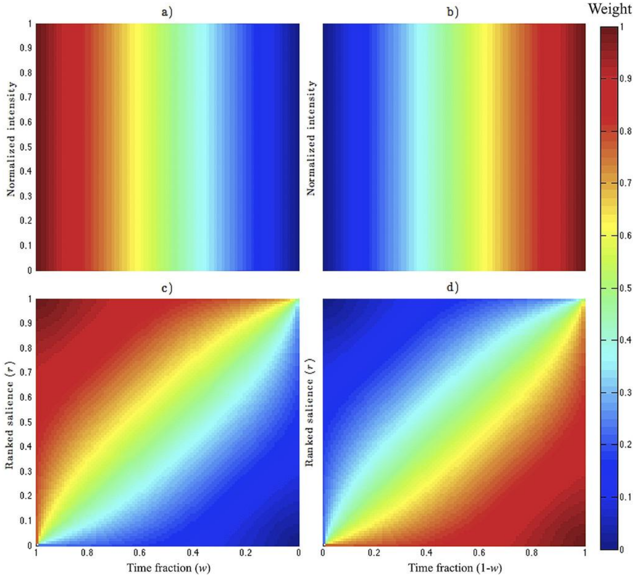


2. Saliency-based blending

- Similar to linear blending, but:
- Preserves pixel intensities over time if they are strong enough according to their ranked salience.

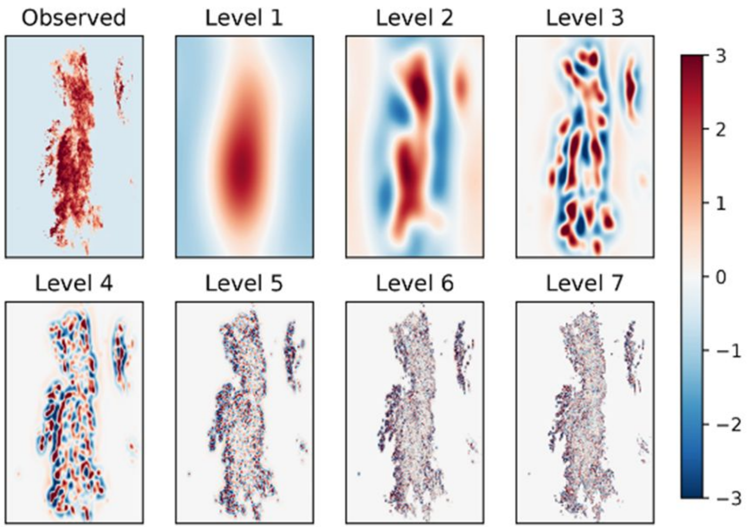
Hwang et al., 2015, Weather and Forecasting

3. STEPS blending (see next slide)



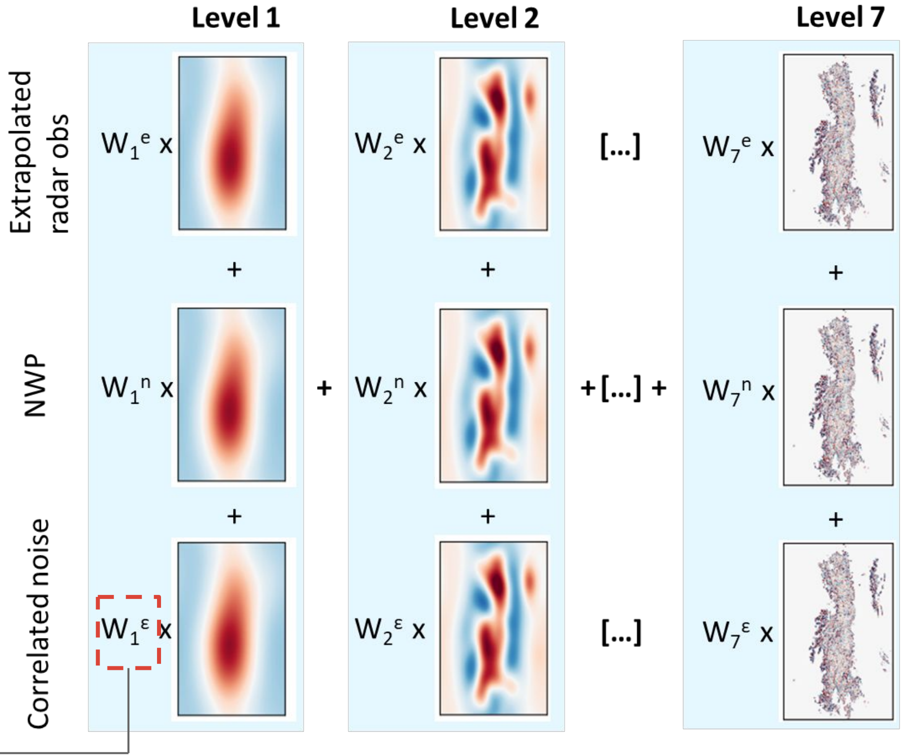
See the QPN session on Monday (16:30 - 18:00) for more information about this method and its' implementation in pysteps!

The STEPS method



Pulkkinen et al., 2019

Per ensemble member:



Blending weights depend on initial and expected future skill