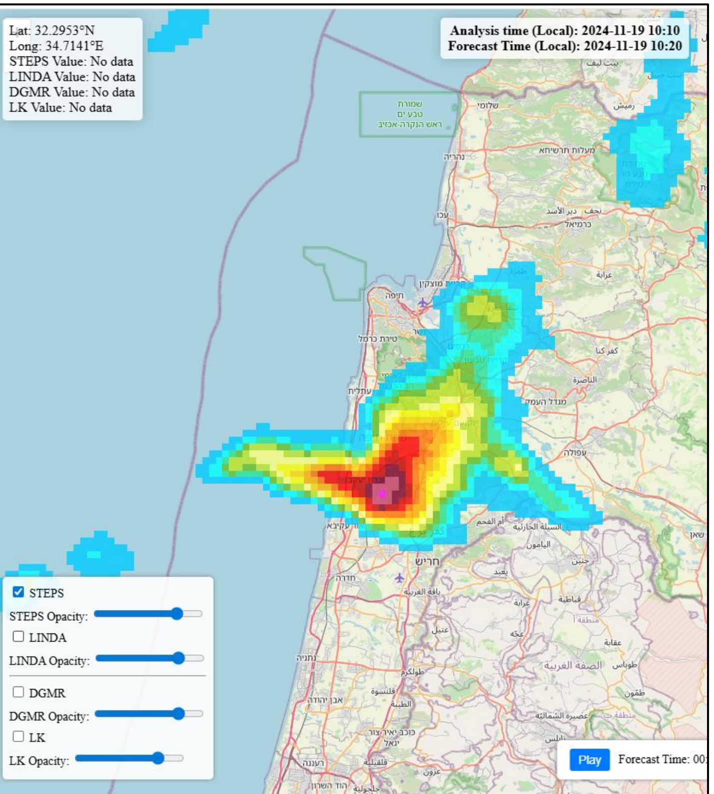
# Radar Nowcasting Methods

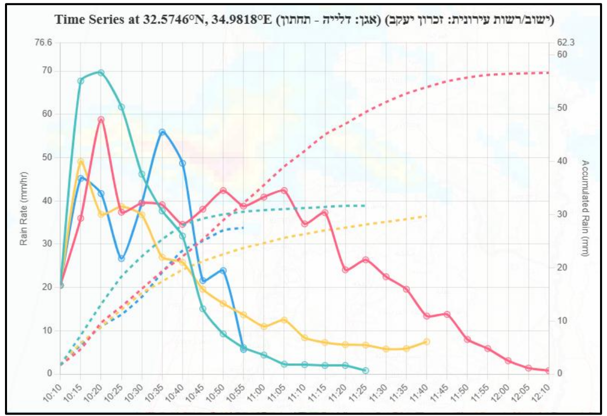
## From Optical Flow to Deep Learning

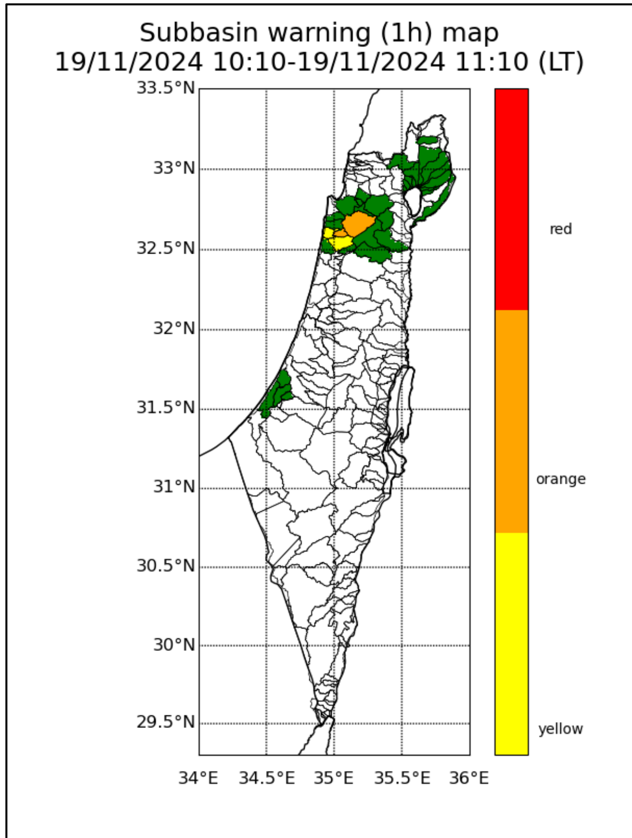

IMS R&D

Elyakom Vadislavsky

vadislavskye@ims.gov.il

# What is Radar Nowcasting?

Nowcasting provides high-resolution forecasts for the very near future (e.g., 0-6 hours).

It is critical for tracking severe weather, like thunderstorms and flash floods, using radar data.

# Outline

- Weather RADAR
- Trec & Optical Flow
- STEPS
- LINDA
- DGMR

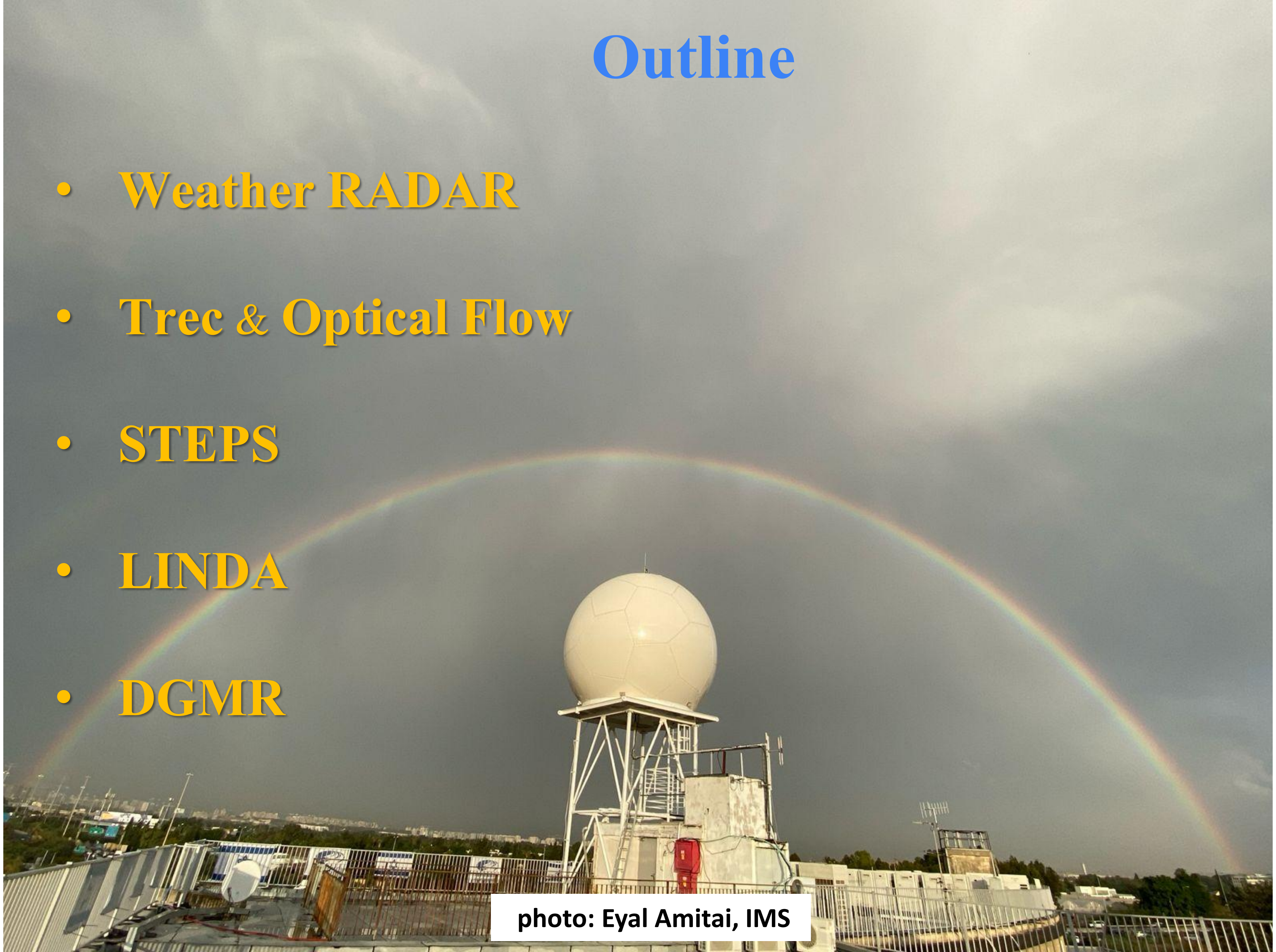
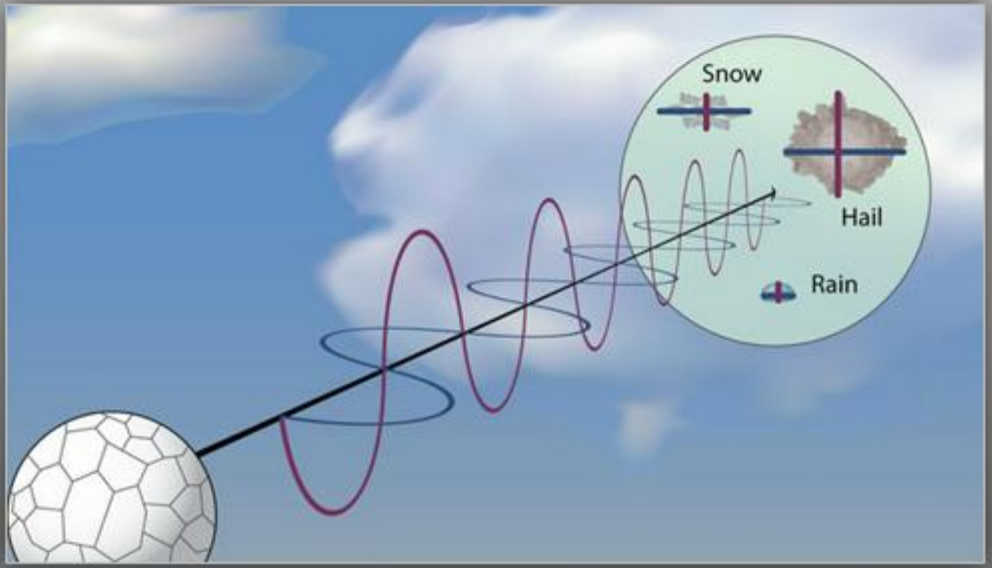


photo: Eyal Amitai, IMS

# Did You Know...

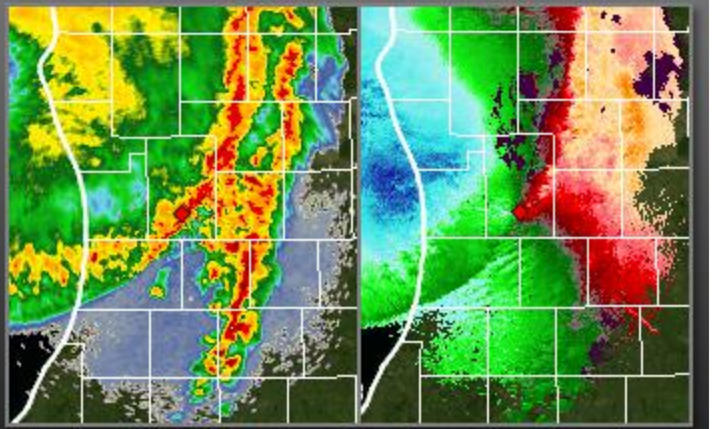
## Radar is short for radio detection and ranging



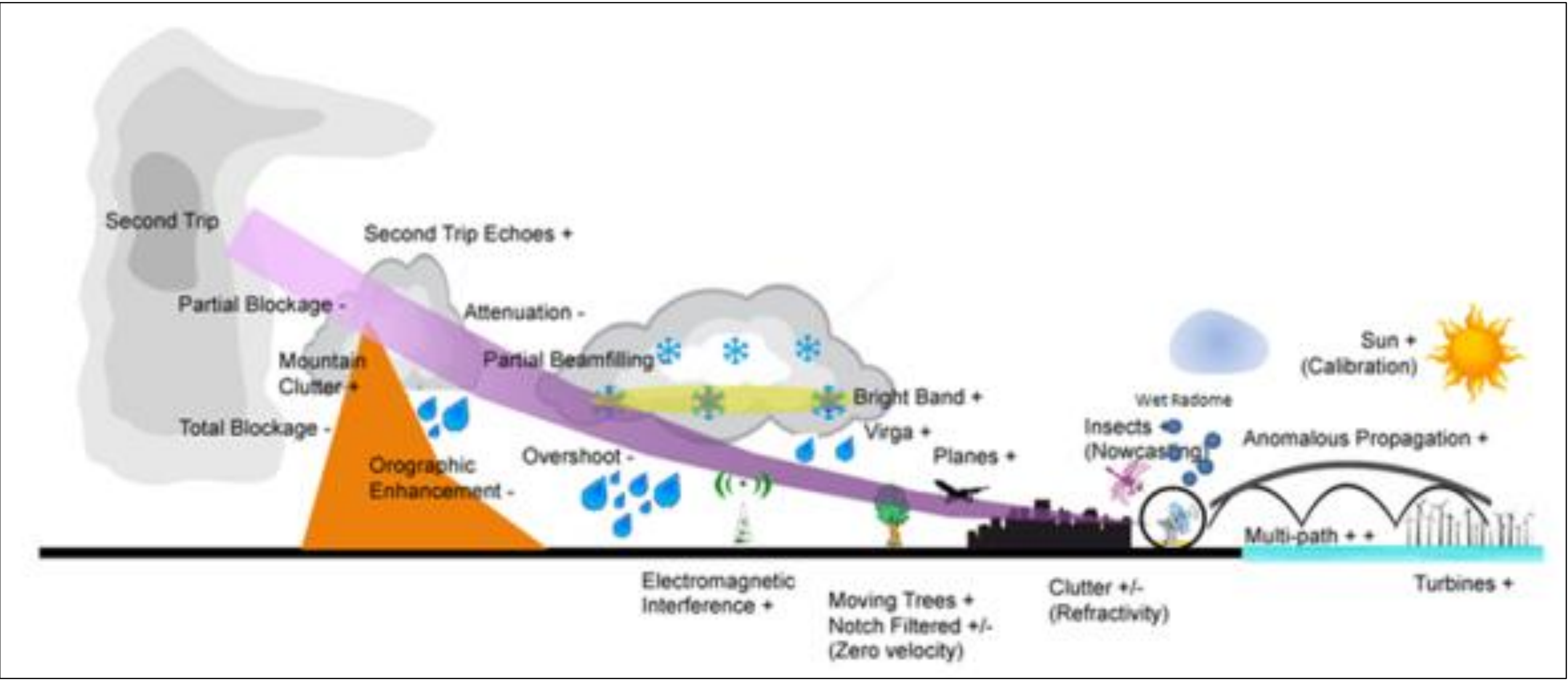

Our Doppler radar emits extremely short bursts of radio waves into the atmosphere, then "listens" for a returning signal

If the energy strikes an object (rain drop, bug, bird, etc.), the energy scatters in all directions and a small fraction of that energy is directed back toward the radar

Precipitation areas and motions toward or away from the radar (Doppler effect) can then be detected



https://www.weather.gov/images/grr/education/dyk/DYK-Monday-RadarFacts.png




https://wmo.int/guide-operational-weather-radar-best-practices-poster

# Cross-Correlation (TREC) & Optical Flow

# TREC Method

## Tracking Radar Echoes by Correlation

The traditional standard for radar echo tracking.

Mechanism: Divides the radar image into small sub-grids or "boxes".

Searches for the most similar box in the subsequent radar scan (Time T+1).

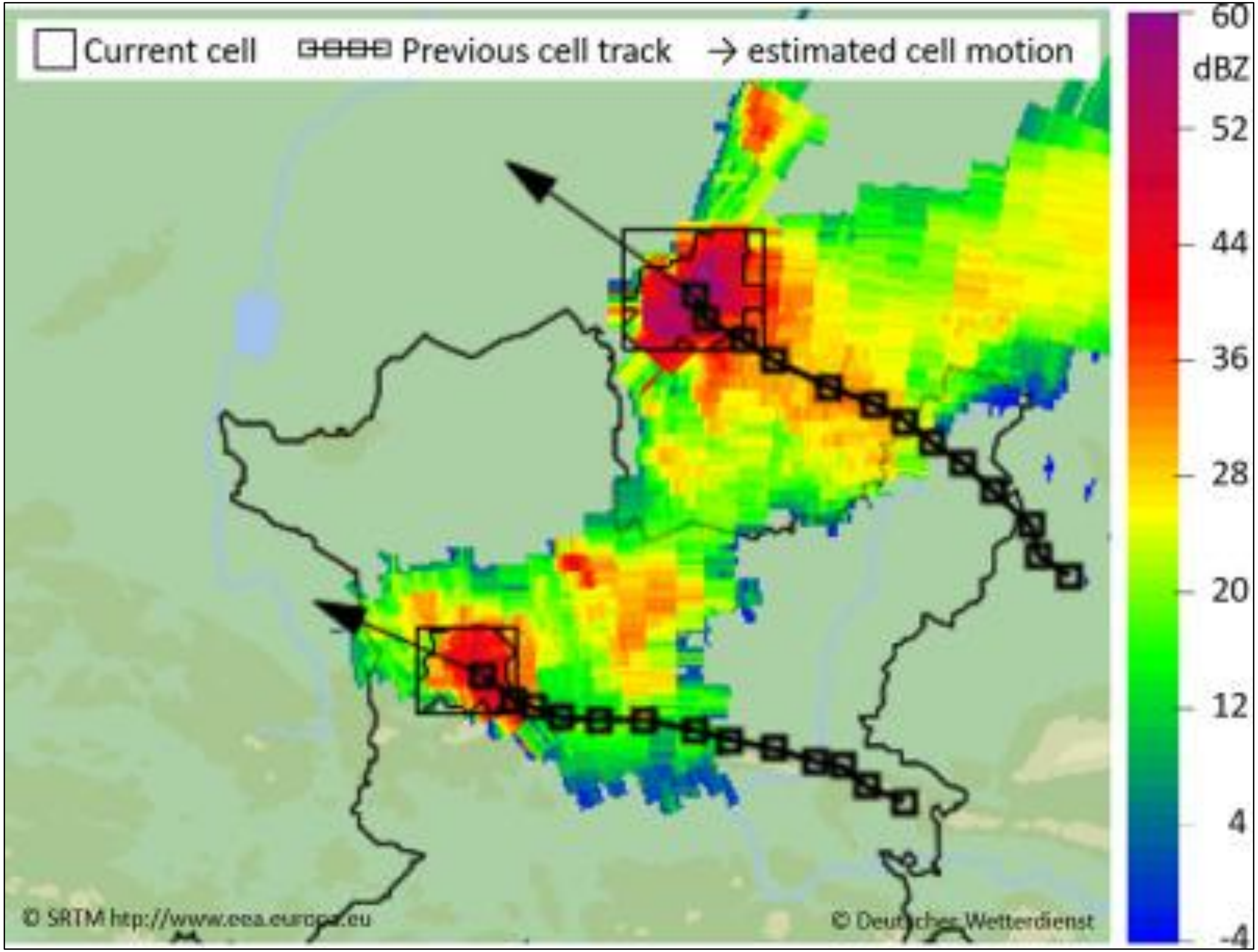
Calculates a motion vector based on the displacement of the best-matched box.

# The Cross-Correlation Coefficient

TREC maximizes the correlation coefficient R between two arrays $Z_1$ (at t) and $Z_2$ (at t+Δt ).

$$R = \frac{\sum (Z_1 - \bar{Z_1})(Z_2 - \bar{Z_2})}{\sqrt{\sum (Z_1 - \bar{Z_1})^2 \sum (Z_2 - \bar{Z_2})^2}}$$

Where Z represents radar reflectivity (dBZ) and $\bar{Z}$ is the mean reflectivity within the search box.




https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/doppler-radar

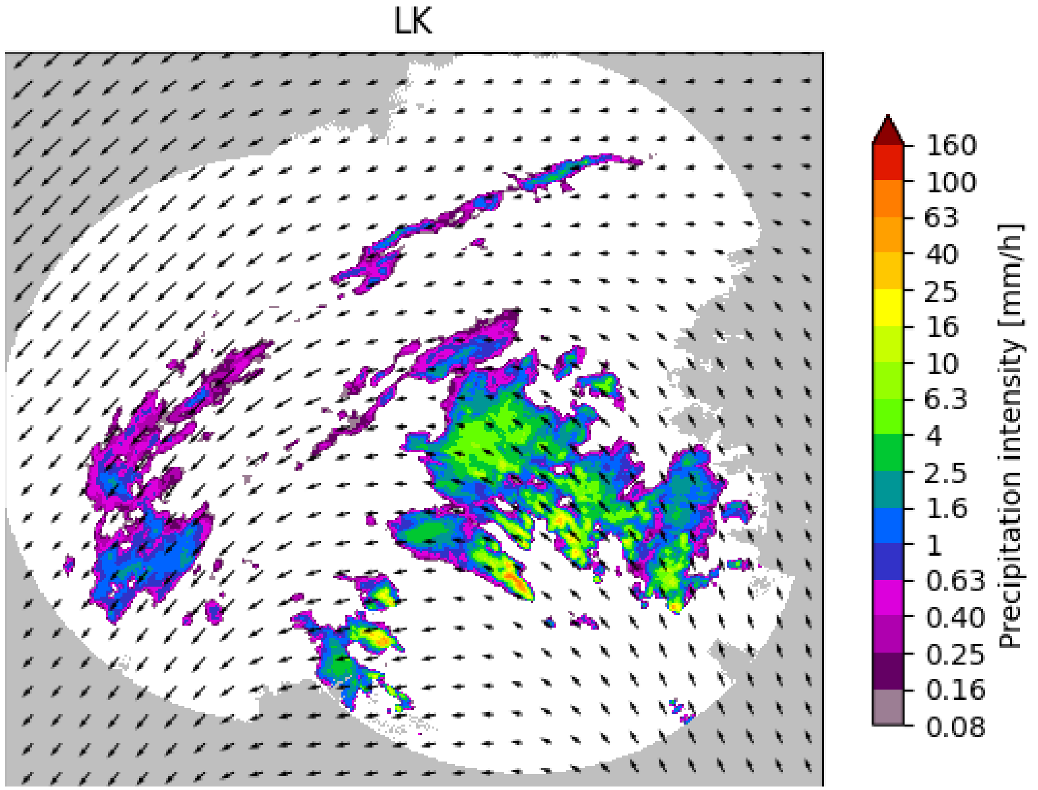
# Optical Flow – Pixel level tracking

Originally from computer vision (Image processing).

Estimates the apparent motion of brightness patterns (reflectivity) at the pixel level.

Produces a dense vector field (a vector for every pixel).

Allows for capturing complex motions like rotation and deformation.




https://pysteps.readthedocs.io/en/latest/auto_examples/plot_optical_flow.html#sphx-glr-auto-examples-plot-optical-flow-py

# Optical Flow Constraint Equation

Brightness constancy assumption - the intensity I of a pixel remains constant as it moves over a short time dt.

$$\frac{\partial I}{\partial t} + u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} = 0$$

$I(x, y, t)$: Radar Reflectivity.

$u, v$: Horizontal and vertical velocity components.

Lucas-Kanade assumes flow is constant in a small local neighborhood. Solves via least squares. Fast but struggles with large displacements.

# Limitations of TREC & Optical Flow

Rigid motion assumption: the shape of the storm cell remains constant within the tracking interval.

Lagrangian persistence: the precipitation fields move with the flow without changing intensity.

# TREC vs. Optical Flow

## Cross-Correlation (TREC)

**Resolution:** Sparse / Block-based (e.g., 1 vector per few km).

**Motion:** Rigid translation of boxes.

**Computation:** Simple, robust to noise but coarse.

**Best For:** Tracking large, stable storm systems.

## Optical Flow

**Resolution:** Dense (1 vector per pixel).

**Motion:** Captures rotation, divergence, and deformation.

**Computation:** More intensive, requires smoothing constraints.

**Best For:** Complex, rapidly evolving convective storms.

# Future Directions

Optical flow is currently the industry standard for short-term (0-2 hour) advection.

Deep Learning (AI)：New models (ConvLSTM, U-Net) are starting to outperform traditional optical flow.

AI can learn non-linear growth and decay, not just motion (advection).

Hybrid systems (Optical Flow+AI) are likely the future of nowcasting.

# References

**Rinehart, R. E.,** & **Garvey, E. T. (1978).** Three-dimensional storm motion detection by conventional weather radar. *Nature*, 273, 287–289.

**Tuttle, J. D.,** & **Foote, G. B. (1990).** Determination of the boundary layer airflow from a single Doppler radar. *Journal of Atmospheric and Oceanic Technology*, 7(2), 218–232.

**Lucas, B. D.,** & **Kanade, T. (1981).** An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, 674–679.

**Bowler, N. E., Pierce, C. E.,** & **Seed, A. (2004).** Development of a precipitation nowcasting algorithm based upon optical flow techniques. *Journal of Hydrology*, 288(1-2), 74–91.

# STEPS

## Short Term Ensemble Prediction System

### A Probabilistic Approach to Precipitation Forecasting

# What is STEPS?

### Definition

STEPS (Short-Term Ensemble Prediction System) is a widely used nowcasting algorithm developed jointly by the UK Met Office and the Australian Bureau of Meteorology.

It bridges the gap between:

- **Radar Extrapolation:** Accurate for 0-60 mins.
- **Numerical Weather Prediction (NWP):** Accurate for >3-6 hours.

**Goal: To generate an ensemble of rainfall cascades that represents the uncertainty in future evolution.**

# Limitations of Simple Extrapolation

Traditional nowcasting uses Lagrangian persistence, moving pixels based on optical flow.

**Small Scales:** Rain cells grow and decay rapidly (predictable only for mins).

**Large Scales:** Frontal systems persist longer (predictable for hours).

*Simple extrapolation fails because it assumes "frozen" turbulence, ignoring the dynamic lifecycle of storm cells.*

# S-PROG (Spectral Prognosis) Model

**Origin:** Introduced by Seed (2003) as a dynamic scaling approach to advection forecasting.

**Core Concept:** Decomposes the precipitation field into a multiplicative cascade of spatial scales (levels).

**Mechanism:** Applies an Auto-Regressive (AR) process to each scale separately to model temporal evolution.

**Function:** Filters out unpredictable, small-scale features to manage forecast uncertainty deterministically.

**Relation:** Serves as the unperturbed baseline for the probabilistic STEPS model

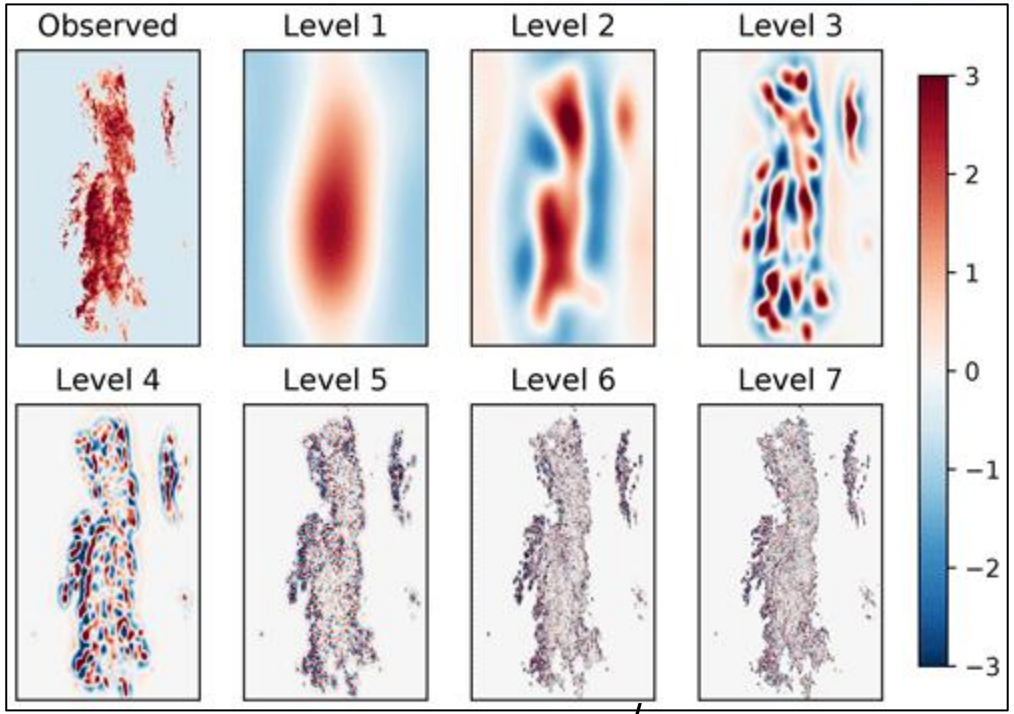
# Scale Decomposition

Lifetime of precipitation relates to its spatial scale.

STEPS decompose the precipitation field into a multiplicative cascade (FFT).

Cascade levels represent different spatial scales.

Each level is treated independently in the forecast.




https://gmd.copernicus.org/articles/12/4185/2019

# STEPS Methodology Flow

### 1. Input

Sequence of recent radar reflectivity fields (converted to dBR).

### 2. Optical Flow

Calculate advection field (motion vectors) using Lucas-Kanade method.

### 3. Evolution

Decompose, advect, and evolve each scale with AR models + Noise.

# The AR model

A typical approach to model temporal evolution of precipitation fields.

Auto-regressive (AR) process, combines the deterministic component from Lagrangian persistence with a stochastic innovation term (noise or perturbation term).

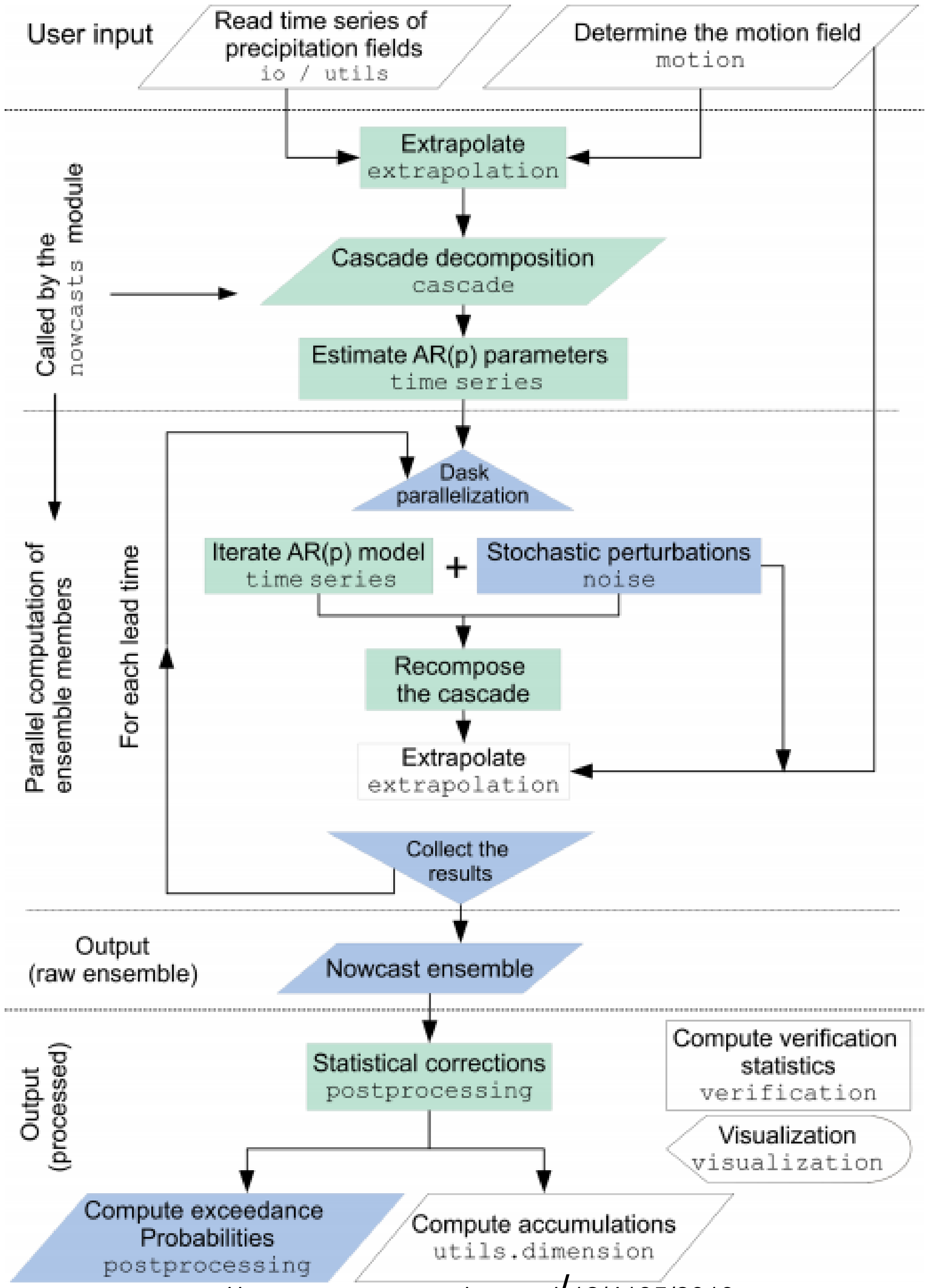
$$R_j(x,y,t) = \sum_{k=1}^{p} \phi_{j,k} R_j(x,y,t-k\Delta t) + \phi_{j,0}\varepsilon_j(x,y,t)$$

https://gmd.copernicus.org/articles//12/4185/2019

**Forecasted rain**

**Deterministic**
Lagrangian persistence

**Stochastic**
Initiation, growth and decay of precipitation




https://gmd.copernicus.org/articles/12/4185/2019

# Stochastic Ensembles

## Why Ensembles?

Because the small scales are unpredictable, STEPS generates multiple "realizations" by adding different random noise seeds.

This results in probabilistic outputs (e.g., "30% chance of >10mm rain") rather than a single deterministic guess.

# References


**Seed, A. W. (2003).** A dynamic and spatial scaling approach to advection forecasting. *Journal of Applied Meteorology*.

**Bowler, N. E., et al. (2006).** STEPS: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled NWP. *Q. J. R. Meteorol. Soc.*

**Pulkkinen, S., et al. (2019).** Pysteps: an open-source Python library for probabilistic precipitation nowcasting. *Geoscientific Model Development*.

**Germann, U., & Zawadzki, I. (2002).** Scale-dependence of the predictability of precipitation from continental radar images. *Monthly Weather Review*.

# LINDA

## Lagrangian INtegro-Difference equation model with Autoregression

# The Challenge: Convective Rainfall

## Limitations of Standard Methods

Traditional nowcasting methods (like simple extrapolation or S-PROG) often fail to capture the rapid **growth and decay** of intense convective storms.

Extrapolation: Assumes "frozen" turbulence; rain cells don't change intensity

S-PROG: Filters small scales to remove unpredictable noise, but this results in "blurred" forecasts for intense local rain.

# What is LINDA?

## Definition

*"Lagrangian INtegro-Difference equation model with Autoregression"*

LINDA model is designed to detect convective rainfall.

Core Philosophy: It treats rainfall evolution as a combination of three distinct physical processes:

1. **Advection:** The motion of the storm.
2. **Growth/Decay:** The change in intensity over time.
3. **Loss of Predictability:** The inevitable uncertainty at small scales.

# The 5 Core Components

### 1. Feature Detection

Identifying rain cells ("blobs") to focus computation on relevant areas.

### 2. Advection

Lagrangian extrapolation using optical flow vectors.

### 3. ARI Process

Autoregressive Integrated model to simulate growth and decay.

### 4. Convolution

Integro-difference equations to model error distribution.

### 5. Stochastic Perturbations

Adding noise to generate probabilistic ensembles.

# 1. Feature Detection (Blobs)

## Targeting Key Features

Unlike global methods that treat the whole domain equally, LINDA uses **feature detection** (e.g., Laplacian of Gaussian "Blob" detector):

- Allows the model to localize parameters to specific storm cells.
- focuses computational resources on high-intensity rainfall areas.
- Improves the estimation of growth and decay for individual cells

# 2. Advection: The Lagrangian Framework

The foundation of the model is the advection equation, solved in Lagrangian coordinates. This describes how the rain field moves with velocity.

$$\frac{d\psi}{dt} = \frac{\partial \psi}{\partial t} + v \cdot \nabla \psi = S(t)$$

**Key Insight:** Standard extrapolation assumes eq = 0. LINDA explicitly models S(t) (the source/sink term) to represent the storm's life cycle.

# 3. Growth & Decay: The ARI Process

To solve for S(t), LINDA uses an Autoregressive Integrated (ARI) process along the Lagrangian trajectory.

This predicts the future intensity based on past states using:

- Autoregressive coefficients (determined empirically).
- Stochastic error term (innovation).

# 4. Predictability Loss & Convolution

## Integro-Difference equation

As forecast lead time increases, small-scale details become unpredictable.

LINDA handles this by applying a **Convolution Kernel** (often anisotropic) to the error term.

This mathematically "smooths" the error distribution, ensuring that uncertain small-scale features don't dominate the forecast, effectively transitioning from a deterministic to a probabilistic view.

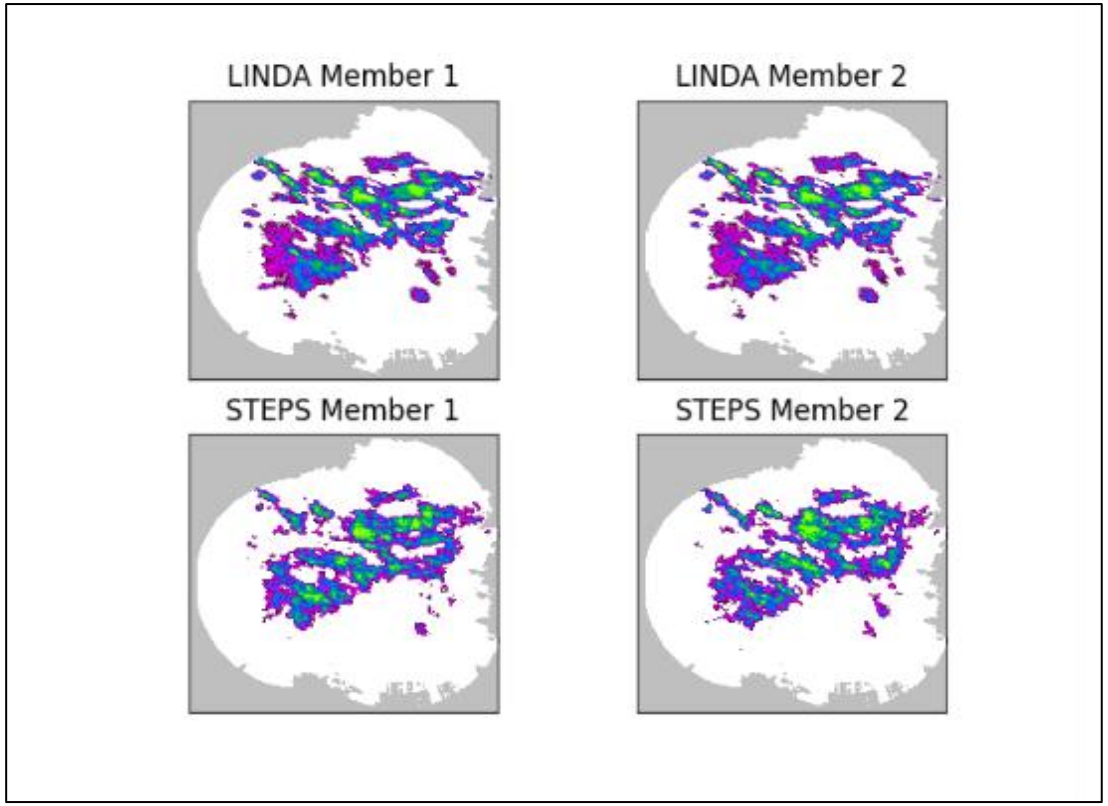
# LINDA vs. STEPS

## Superiority in Convection

**STEPS/S-PROG:** Uses spectral decomposition. It tends to filter out high frequencies, leading to "blurred" forecasts for intense storms.

**LINDA:** Preserves local structure better due to the Lagrangian ARI process.

**Result:** LINDA produces sharper, more structurally accurate nowcasts for heavy, localized rainfall events.




https://pysteps.readthedocs.io/en/latest/auto_examples/linda_nowcasts.html

# References

**Pulkkinen, S., et al. (2021).** "Lagrangian Integro-Difference Equation Model for Precipitation Nowcasting." *Journal of Atmospheric and Oceanic Technology*, 38(12).

**Pulkkinen, S., et al. (2019).** "Pysteps: an open-source Python library for probabilistic precipitation nowcasting." *Geoscientific Model Development*, 12.

**Germann, U.**, & **Zawadzki, I. (2002).** "Scale-dependence of the predictability of precipitation from continental radar images." *Monthly Weather Review*, 130.

**Seed, A. W. (2003).** "A dynamic and spatial scaling approach to advection forecasting." *Journal of Applied Meteorology*.

# DGMR

## Deep Generative Model of Rainfall

DeepMind & Met Office

Collaboration

# The Challenge of Nowcasting

**Goal:** High-resolution precipitation forecasting up to 90 minutes ahead.

**Traditional Methods:** Rely on advection (optical flow). Good for movement, bad for intensity changes or new formation.

**Deep Learning:** Uses Mean Squared Error (MSE) loss. Result is often *blurry* predictions that fail to capture extreme events.

# DGMR Overview

## Probabilistic Generative Approach

DGMR moves beyond deterministic regression.

**Generative:** Produces multiple plausible future realizations (samples) rather than one blurry average.

**Adversarial:** Uses a GAN (Generative Adversarial Network) framework to ensure realism.

**Core Idea:** Balance the *sharpness* of GANs with the *consistency* of regularization

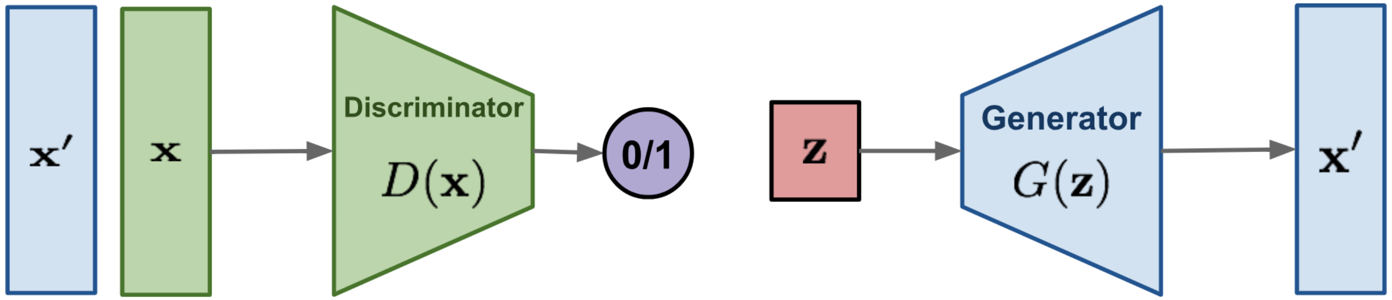
# Generator Architecture

**Conditioning Stack:** Processes past 4 radar frames using residual D-Blocks.

**Latent Stack:** Injects Gaussian noise ($Z$) to allow for probabilistic diversity.

**Sampler:** A ConvGRU (Convolutional Gated Recurrent Unit) network that recurrently predicts 18 future frames (90 mins).

**GAN:** minimax the classification error loss.




https://lilianweng.github.io/posts//sledom-wofl-2018-10-13

# Dual Discriminators

## Spatial Discriminator

A Convolutional Neural Network (CNN) that ensures each individual frame looks like a realistic radar field. It focuses on spatial texture and intensity distributions.

## Temporal Discriminator

A 3D-CNN that examines sequences of frames. It ensures the motion is fluid and temporally consistent, penalizing "jumpy" or unrealistic evolutions.

# Equations I: Objective Function

The model uses a **Hinge Loss** formulation for the adversarial component. This is known to provide stable gradients for training.

**Discriminator Loss ($L_D$)**

$$L_D = E_{x \sim p_{data}} [ \max(0, 1 - D(x)) ] + E_{z \sim p_z} [ \max(0, 1 + D(G(z))) ]$$

**D**: Discriminator

**G**: Generator

**x**: Real Data

**z**: Latent Noise

# Equations II: Regularization

## Grid Cell Regularization

To make predictions more realistic, a regularization term lambda*$L_{reg}$ is added. This term penalizes the difference between the generated samples and the ground truth, often heavily weighting high-intensity rainfall pixels to prevent "missing" extreme events.

**Generator Loss ($L_G$)**

$$L_G = -E_{z \sim p_z}[D(G(z))] + \lambda L_{reg}$$

# Performance Comparison

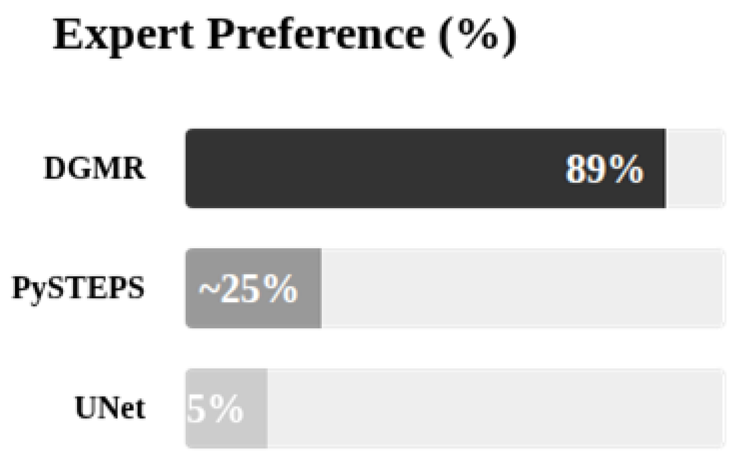
## Superior Results

In a cognitive evaluation with over 50 expert meteorologists from the UK Met Office, DGMR was significantly preferred.

**Accuracy:** Better spatial extent of rain.

**Utility:** More useful for flood warnings.

**No Blurring:** Maintains crisp details.



# Visual Results

## Sharp vs. Blurry

The visual difference is striking. Deterministic models (like UNet) average out uncertainties, leading to a "foggy" look that underestimates peak rainfall intensity.

DGMR maintains the "sharpness" and texture of real radar data, preserving the small-scale convective features crucial for flash flood warnings.

# References

Ravuri, S., Lenc, K., Willson, M. et al. **Skillful precipitation nowcasting using deep generative models of radar.** *Nature* 597, 672–677 (2021).

Pulkkinen, S. et al. **Pysteps: an open-source Python library for probabilistic precipitation nowcasting.** *Geosci. Model Dev.* 12, 4185–4219 (2019).

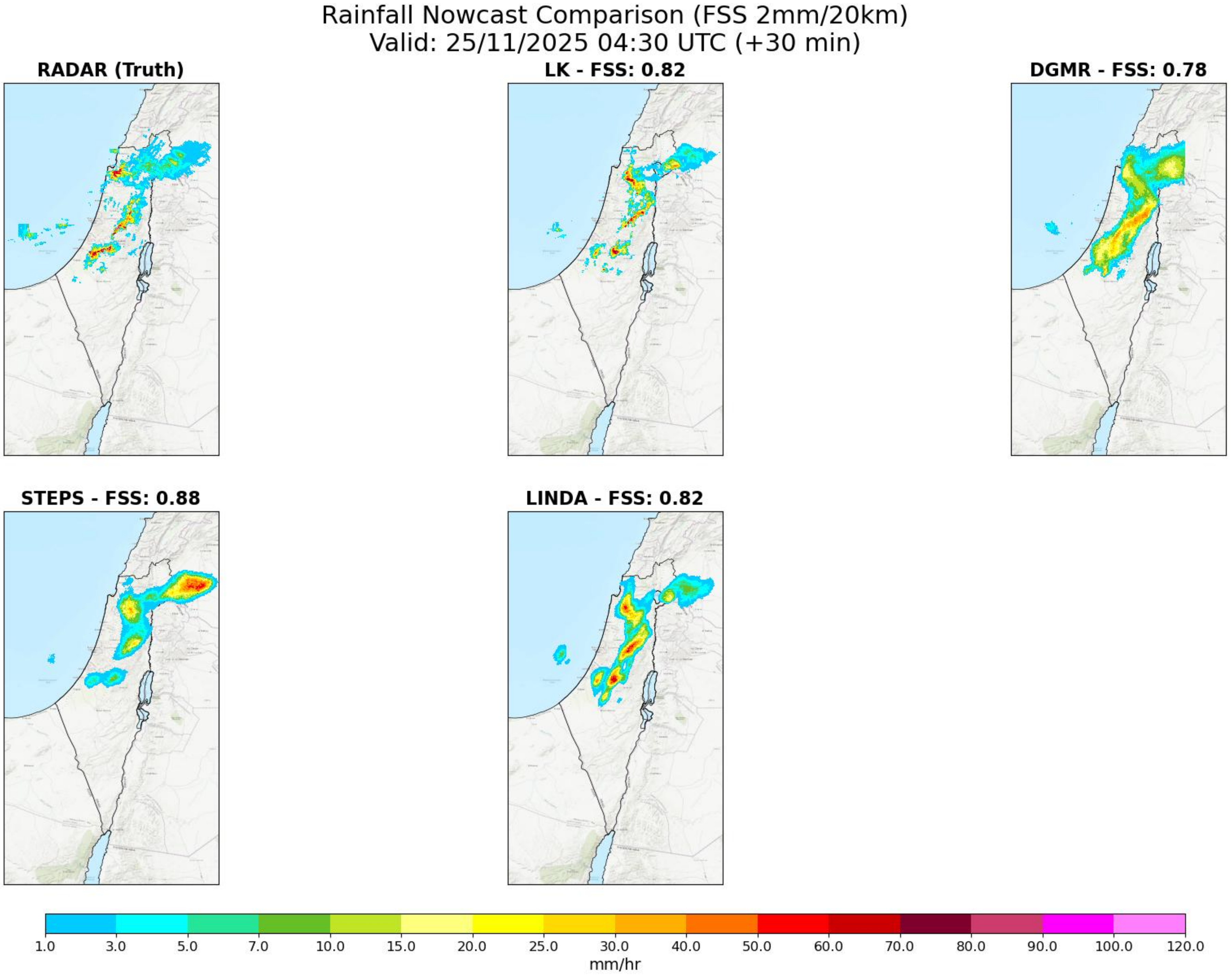
Goodfellow, I. et al. **Generative adversarial nets.** *Adv. Neural Inf. Process. Syst.* 27 (2014).

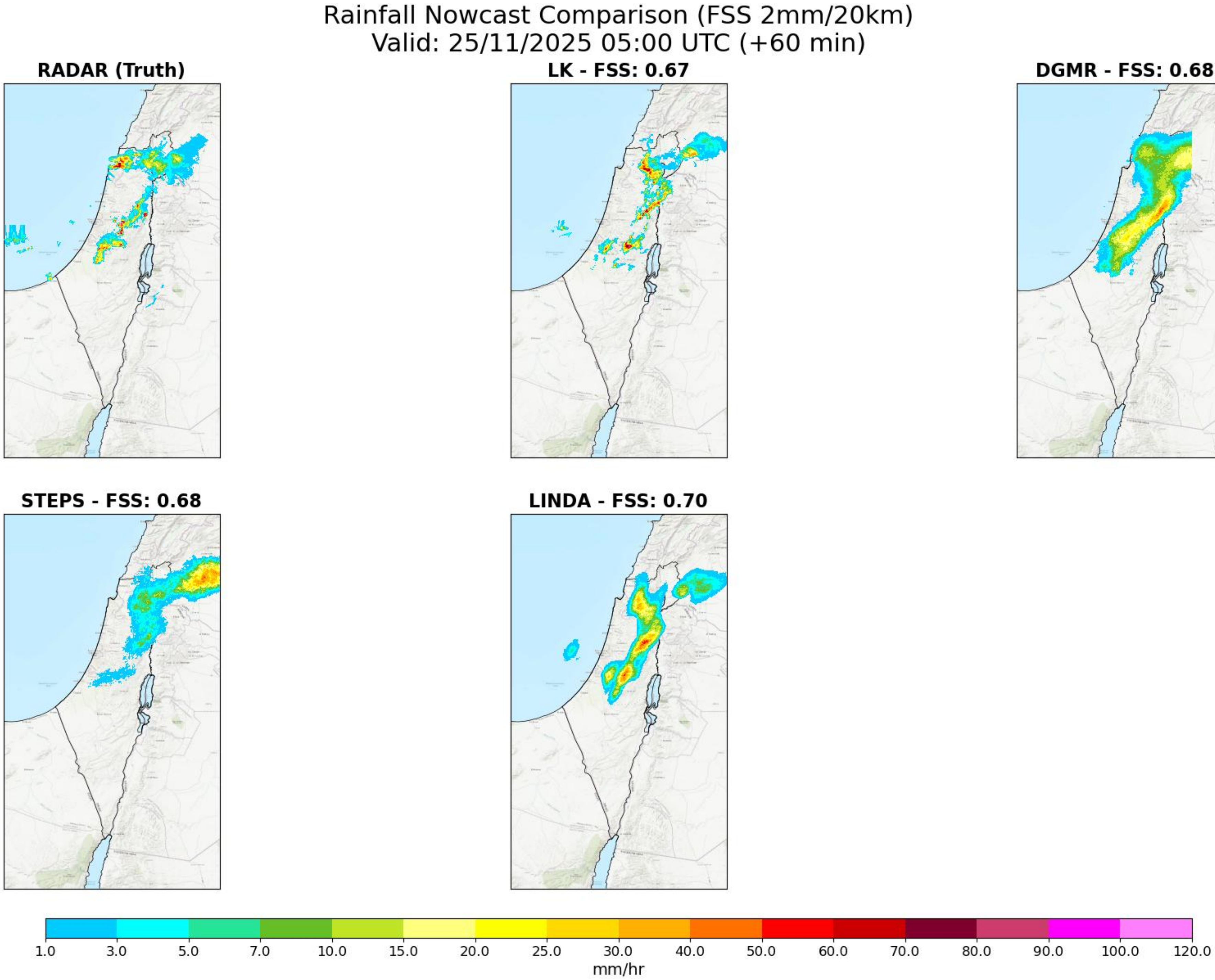
Shi, X. et al. **Convolutional LSTM network: A machine learning approach for precipitation nowcasting.** *Adv. Neural Inf. Process. Syst.* (2015).

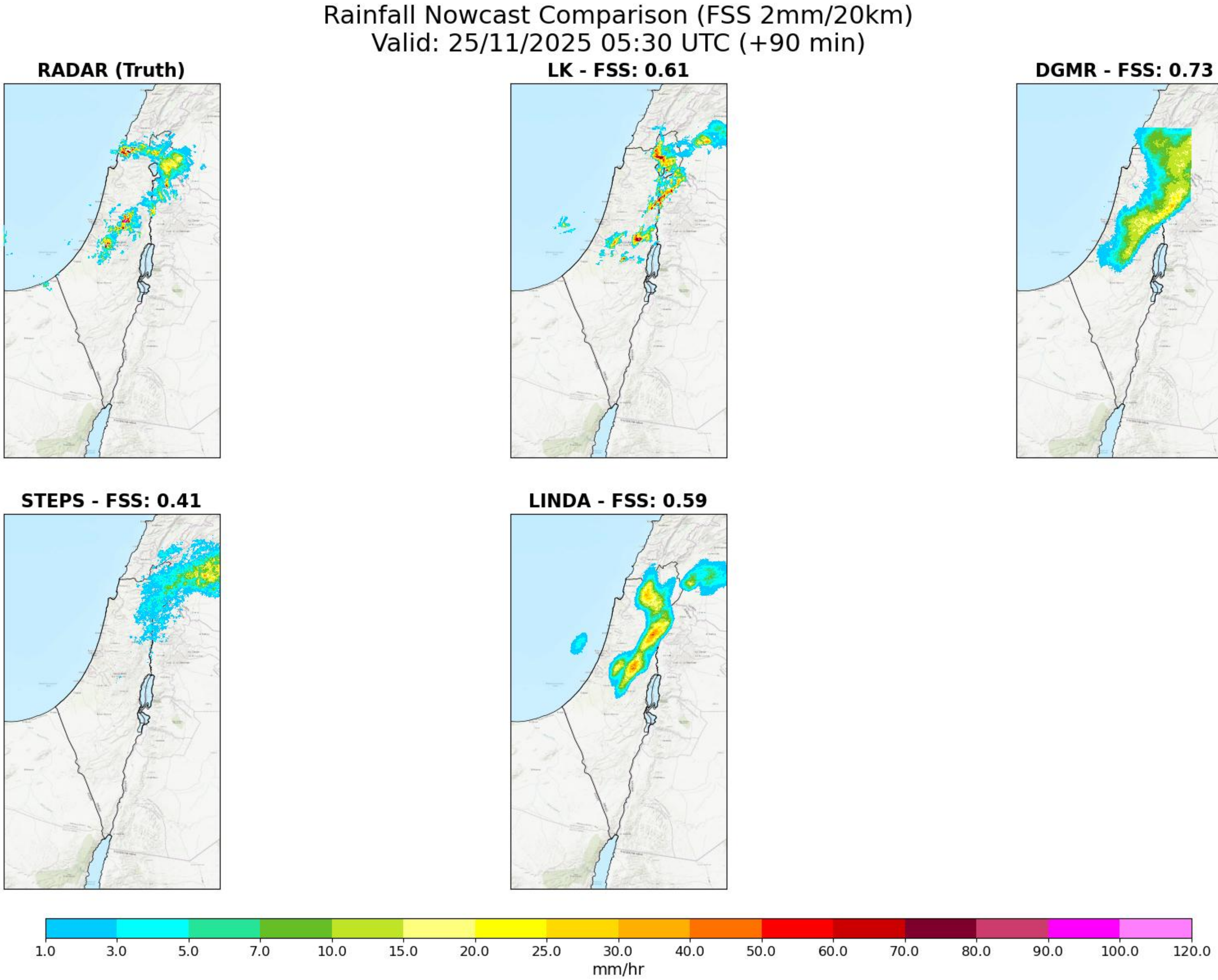
# Available Radar Nowcasting Tools

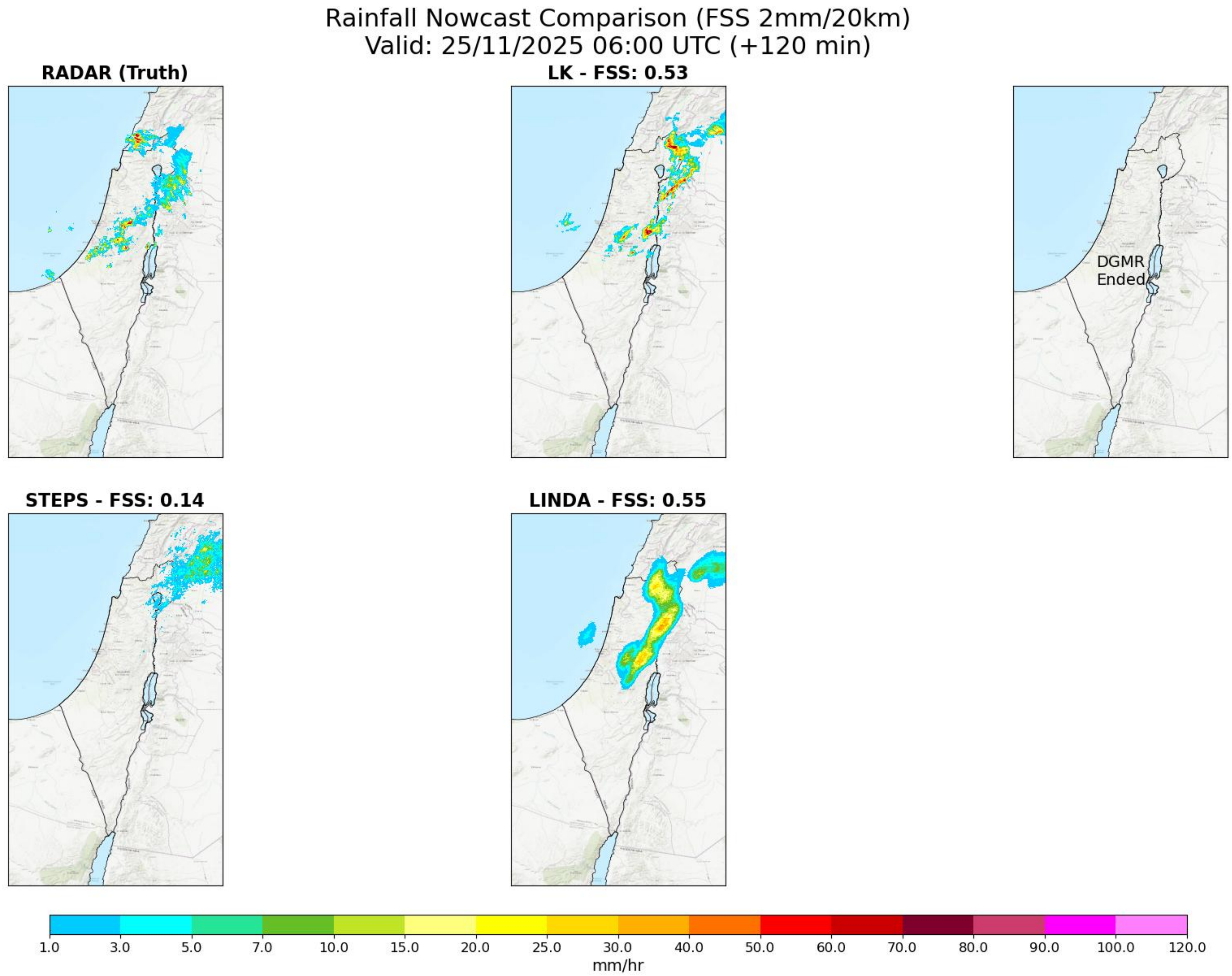
| Tool | Methodology | Key Features | Availability | Website | Reference |
|---|---|---|---|---|---|
| INCA | Sensor Fusion | Alpine terrain correction; integrates stations & sat. | Operational / Commercial | www.geosphere.at | Haiden et al. (2011) |
| pySTEPS | Optical Flow | Modular, probabilistic, extensive library. | Open Source (BSD-3) | pysteps.github.io | Pulkkinen et al. (2019) |
| DGMR | Deep Gen (GAN) | Generative models, sharp predictions. | Open Source (Apache) | github/deepmind | Ravuri et al. (2021) |
| TITAN | Centroid Tracking | Object-based storm cell tracking. | Open Source (BSD) | www.lrose.net | Dixon & Wiener (1993) |
| SWIRLS | Semi-Lagrangian | "Com-SWIRLS" version available. | Restricted / Community | swirls.hko.gov.hk | Li & Lai (2004) |
| MetNet-3 | Neural Weather | 24h horizon, 2min resolution. | Proprietary (API) | research.google | Sonderby et al. (2020) |
| Rainymotion | Dense Flow | Lightweight tracking & extrapolation. | Open Source (MIT) | github/hydrogo | Ayzel et al. (2019) |

# Example of 4 methods:
# 24-25/11/2025 Severe weather event


Rainfall Nowcast Comparison (FSS 2mm/20km)
Valid: 25/11/2025 04:30 UTC (+30 min)
RADAR (Truth)
LK - FSS: 0.82
DGMR - FSS: 0.78
STEPS - FSS: 0.88
LINDA - FSS: 0.82
1.0
3.0
5.0
7.0
10.0
15.0
20.0
25.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
120.0
mm/hr


Rainfall Nowcast Comparison (FSS 2mm/20km)
Valid: 25/11/2025 05:00 UTC (+60 min)
RADAR (Truth)
LK - FSS: 0.67
DGMR - FSS: 0.68
STEPS - FSS: 0.68
LINDA - FSS: 0.70
1.0
3.0
5.0
7.0
10.0
15.0
20.0
25.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
120.0
mm/hr


Rainfall Nowcast Comparison (FSS 2mm/20km)
Valid: 25/11/2025 05:30 UTC (+90 min)
RADAR (Truth)
LK - FSS: 0.61
DGMR - FSS: 0.73
STEPS - FSS: 0.41
LINDA - FSS: 0.59
1.0
3.0
5.0
7.0
10.0
15.0
20.0
25.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
120.0
mm/hr


Rainfall Nowcast Comparison (FSS 2mm/20km)
Valid: 25/11/2025 06:00 UTC (+120 min)
RADAR (Truth)
LK - FSS: 0.53
DGMR
Ended
STEPS - FSS: 0.14
LINDA - FSS: 0.55
1.0
3.0
5.0
7.0
10.0
15.0
20.0
25.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
120.0
mm/hr

# *Thank you for listening*