

# Spatial Verification Techniques for Extreme Events: A Review

Carlos Peralta

2025-07-16

## Introduction

Brief overview of the “double penalty problem” in traditional gridpoint-based verification and why spatial verification techniques are needed for high-resolution forecasts of extreme events.

## The Double Penalty Problem

- Traditional metrics penalize forecasts twice for spatial displacement
- Example: correct intensity/timing but wrong location → poor scores
- Need for spatial verification methods that don't require exact gridpoint matching

## Classification of Spatial Verification Techniques

Based on Gilleland et al. (2009), spatial verification techniques fall into four main categories:

### 1. Fuzzy/Neighborhood Verification

- **Concept:** Require approximate agreement in space, time, intensity
- **Approach:** Measure strength of agreement as closeness requirements vary
- **Examples:**
  - Neighborhood verification methods
  - Fractions Skill Score (FSS)
- **Pros:**
  - Intuitive and easy to implement

- **Cons:**
  - May lose information about specific displacement patterns

## 2. Scale-Decomposition Techniques

- **Concept:** Assess forecast skill at different spatial scales by filtering the fields.
- **Approach:** Apply spatial filters (e.g., Fourier, wavelet) to separate and analyze features by scale.
- **Examples:**
  - Intensity-Scale technique (Casati et al. 2004)
  - Wavelet-based verification methods
- **Pros:**
  - Reveals at which scales the model performs well or poorly
  - Helps isolate errors in large-scale vs. small-scale features
- **Cons:**
  - Can be complex to interpret and implement
  - May not directly address feature displacement

## 3. Feature-Based/Object-Oriented Techniques

- **Concept:** Identify and compare discrete weather features (objects) in forecasts and observations.
- **Approach:** Detect, match, and analyze properties of corresponding features (e.g., location, size, intensity).
- **Examples:**
  - Ebert & McBride (2000) object-based method
  - MODE (Method for Object-based Diagnostic Evaluation, Davis et al. 2006)
  - SAL (Structure, Amplitude and Location, Wernli and Paulat 2008)
- **Pros:**
  - Intuitive and effective for well-defined features
  - Provides detailed diagnostics on feature attributes
- **Cons:**
  - Challenging to define and match objects, especially in complex fields
  - May struggle with overlapping or poorly defined features

## 4. Field Verification Techniques (Optical Flow)

- **Concept:** Compare entire spatial fields directly, without decomposing into objects or scales.
- **Approach:** Use algorithms (e.g., optical flow) to morph the forecast field to match the observation, quantifying displacement and amplitude errors.
- **Examples:**
  - Displacement and Amplitude Score (DAS, Keil & Craig 2009)
  - Other optical flow-based methods
- **Pros:**
  - Avoids need for explicit feature identification or matching
  - Captures both displacement and amplitude errors in a unified framework
- **Cons:**
  - Sensitive to algorithm parameters (e.g., search distance)
  - Computationally intensive and may be less interpretable for complex cases

## Examples of spatial verification scores with focus on weather extremes

### 1. Fuzzy/Neighborhood Verification

- **Structure of Local Extremes (SLX):**

When it comes to precipitation **extremes**, typically what we want to know is

- Where will the **heaviest rain** fall?
- Where will it stay **completely dry**?

**SLX (Structure of Local Extremes)** by Sass (2021) evaluates the capability of high resolution models to predict extremes by using neighbourhood verification focused specifically on extremes. Two scores identify how well forecasts identify analysed maxima and minima in the model domain and two other scores identifies how well analysed values identify forecasted maxima and minima.

### The SLX Method

SLX computes four neighbourhood-based scores:

Component	What it measures
SLX_ob_max	How well a forecast captures observed maxima locations
SLX_fc_max	How well an observed field captures forecast maxima locations
SLX_ob_min	How well a forecast captures observed minima locations
SLX_fc_min	How well an observed field captures forecast minima locations

$$SLX = \frac{1}{4}(SLX_{ob\_max} + SLX_{fc\_max} + SLX_{ob\_min} + SLX_{fc\_min})$$

## Complete SLX Algorithm

### Step 1: Domain Setup and Boundary Zones

- **Input:** Forecast field  $\Psi(i,j)$  and analysis field  $\Psi(i,j)$  with same horizontal resolution
- **Boundary zone:** Width  $L\_max$  around domain edges to ensure consistent computations
- **Sub-domains:** Large domains ( $>1000$  grid points) can be partitioned into sub-domains
- **Internal points:** Only internal points (excluding boundary zone) are used for extreme identification

### Step 2: Extreme Point Identification

**Tolerance parameter** ( $kg/m^2$ ): Reflects analysis uncertainty level

- **Default:**  $0.1 kg/m^2$  (close to zero)
- **Alternative:**  $= 5-10\%$  of extreme value for coherent groups
- **Zero areas:** Multiple zero-valued points automatically selected as minima

**Four types of extremes identified:**

1. **obmax(K1):** Observed local maxima (M1 points total)
2. **obmin(K2):** Observed local minima (M2 points total)
3. **fcmax(K3):** Forecast local maxima (M3 points total)

4. **fmin(K4)**: Forecast local minima (M4 points total)

### Step 3: Neighborhood Definition

**Square neighborhoods** around each extreme point: - **Width L**: Measured in grid points (L = 0 means single grid point) - **Size**:  $(2L + 1)^2$  total grid points in neighborhood - **Range**: Test multiple L values from 0 to L\_max

### Step 4: Neighborhood Value Extraction

For each extreme point, find corresponding value in other field's neighborhood:

**Observed maxima → Forecast neighborhood maxima:**

$$\phi_{\max}(L, K1) = \max\{\phi(i, j)\}, \quad i \in [i_{K1} - L, i_{K1} + L], j \in [j_{K1} - L, j_{K1} + L]$$

**Observed minima → Forecast neighborhood minima:**

$$\phi_{\min}(L, K2) = \min\{\phi(i, j)\}, \quad i \in [i_{K2} - L, i_{K2} + L], j \in [j_{K2} - L, j_{K2} + L]$$

**Forecast maxima → Analysis neighborhood maxima:**

$$\Psi_{\max}(L, K3) = \max\{\Psi(i, j)\}, \quad i \in [i_{K3} - L, i_{K3} + L], j \in [j_{K3} - L, j_{K3} + L]$$

**Forecast minima → Analysis neighborhood minima:**

$$\Psi_{\min}(L, K4) = \min\{\Psi(i, j)\}, \quad i \in [i_{K4} - L, i_{K4} + L], j \in [j_{K4} - L, j_{K4} + L]$$

### Step 5: Score Function Application

The **piecewise linear score function**  $S(\phi, \text{ob})$  compares forecast ( $\phi$ ) to observation (ob):

$$S(\phi, \text{ob}) = \begin{cases} \frac{\phi}{\text{ob} - k}, & \phi < \text{ob} - k \\ 1, & \text{ob} - k \leq \phi \leq \text{ob} \\ \max\left(1 - \frac{\phi - \text{ob}}{A \cdot \text{ob}}, 0\right), & \phi > \text{ob} \end{cases}$$

**For small observed values** ( $\text{ob} \leq k$ ):

$$S(\phi, \text{ob}) = \begin{cases} 1, & \phi \leq k \\ \max\left(1 - \frac{\phi - k}{A \cdot k}, 0\right), & \phi > k \end{cases}$$

**Parameters:** -  $k = 0.1$  mm (uncertainty threshold) -  $A = 4$  (asymmetry factor - penalizes over-forecasting by factor  $>5$ )

### Step 6: Individual Score Computation

**SLX\_ob\_max:** How well forecast captures observed maxima

$$SLX_{ob\_max} = \frac{1}{M1} \sum_{K1=1}^{M1} S(\phi_{\max}(L, K1), obmax(K1))$$

**SLX\_ob\_min:** How well forecast captures observed minima

$$SLX_{ob\_min} = \frac{1}{M2} \sum_{K2=1}^{M2} S(\phi_{\min}(L, K2), obmin(K2))$$

**SLX\_fc\_max:** How well analysis captures forecast maxima

$$SLX_{fc\_max} = \frac{1}{M3} \sum_{K3=1}^{M3} S(\Psi_{\max}(L, K3), fcmax(K3))$$

**SLX\_fc\_min:** How well analysis captures forecast minima

$$SLX_{fc\_min} = \frac{1}{M4} \sum_{K4=1}^{M4} S(\Psi_{\min}(L, K4), fcmin(K4))$$

### Step 7: Final SLX Score

$$SLX = \frac{1}{4} (SLX_{ob\_max} + SLX_{fc\_max} + SLX_{ob\_min} + SLX_{fc\_min})$$

### Algorithm Implementation Details

#### Computational Steps (for SLX\_ob\_max component):

1. **Choose neighborhood widths:** Select L values to test (e.g., L = 0, 1, 2, ..., L\_max)
2. **Identify observed maxima:** Find obmax(K1) positions and values
3. **Extract forecast neighborhood maxima:** Compute \_max(L,K1) for each L
4. **Apply score function:** Calculate S( \_max(L,K1), obmax(K1)) for each extreme
5. **Average scores:** Compute mean over all M1 extreme points
6. **Repeat for all L:** Generate SLX\_ob\_max(L) curve

## Key Algorithm Properties

- **Asymmetric scoring:** Penalizes over-forecasting more than under-forecasting
- **Scale-dependent:** Performance varies with neighborhood size  $L$
- **Handles zero fields:** Works with completely dry or wet conditions
- **Operational ready:** Produces daily scores for any input fields
- **Flexible domains:** Supports sub-domain analysis for large areas

## Operational Considerations

- **Execution time:** Few minutes for domains  $\sim 200 \times 200$  grid points
- **Implementation:** Coded in R programming language
- **Memory optimization:** Can use subset of extreme points for large domains
- **Daily output:** Designed for routine operational verification
- **Post-processing:** Can extract statistics for specific precipitation thresholds

## Visual Explanation of SLX Algorithm

The SLX algorithm works by:

1. **Identifying extremes** in both forecast and observed fields
2. **Creating square neighborhoods** around each extreme point
3. **Finding corresponding extremes** in the other field's neighborhood
4. **Scoring the match quality** using the piecewise linear function
5. **Averaging scores** across all extreme points and components

**Critical insight:** The neighborhood size  $L$  determines the spatial scale at which extremes are successfully captured. Larger  $L$  values are more forgiving of displacement errors but may miss fine-scale structure.

## Score Function Properties

- **Perfect match**  $\rightarrow S = 1$
- **Severe over-forecast** ( $>5\times$ )  $\rightarrow S = 0$
- **Asymmetric:** Designed to avoid under-forecasting of warning conditions
- **Piecewise linear:** Simple but effective for operational use
- **Uncertainty aware:**  $k$  parameter accounts for observation uncertainty

## Advantages of SLX

- **Extreme-focused:** Specifically designed for precipitation extremes
- **Neighborhood-based:** Addresses double penalty problem
- **Comprehensive:** Evaluates both maxima and minima
- **Scale-aware:** Tests multiple neighborhood sizes
- **Operationally practical:** Fast computation, daily output
- **Flexible:** Adaptable score function for different applications

## Limitations of SLX

- **Parameter sensitivity:** Results depend on  $k$ ,  $A$ ,  $\epsilon$ , and  $L$  choices
- **Score function complexity:** Piecewise linear function may need refinement
- **Extreme definition:** Tolerance parameter  $\epsilon$  affects extreme selection
- **Computational scaling:** May need optimization for very large domains