

Supplement of Geosci. Model Dev., 12, 1387–1402, 2019
<https://doi.org/10.5194/gmd-12-1387-2019-supplement>
© Author(s) 2019. This work is distributed under
the Creative Commons Attribution 4.0 License.



Supplement of

Optical flow models as an open benchmark for radar-based precipitation nowcasting (rainymotion v0.1)

G. Ayzel et al.

Correspondence to: Georgy Ayzel (ayzel@uni-potsdam.de)

The copyright of individual parts of the supplement might differ from the CC BY 4.0 License.

S1 Optical flow models' parameters

S1.1 The Sparse group

Shi–Tomasi corner detector parameters:

- Maximum number of corners to return (default: 200; ranges: [50, 300])
- 5 – Minimal accepted quality of image corners (default: 0.2; ranges: [0.1, 0.7])
- Minimum possible Euclidean distance between the returned corners (default: 0.2; ranges: [0.1, 0.7])
- Size of an average block for computing a derivative covariation matrix over each pixel neighborhood (default: 21; ranges: [10, 50])

Lucas–Kanade tracking algorithm:

- 10 – Size of the search window at each pyramid level (default: 20; ranges: [10, 30])
- Zero-based maximal pyramid level number (default: 2; ranges: [0, 4])

Interpolation algorithm:

- Regression model for tracking features extrapolation (default: linear regression)
- Features are used for building a regression model (default: ordinal, options: [polynomial])
- 15 – Warping scheme (default: affine, options: [euclidean, similarity, projective])

Parameters values were set up based on trial-and-error experimentation and the following online sources:

- http://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=goodfeaturestotrack#goodfeaturestotrack
- http://docs.opencv.org/2.4/modules/video/doc/motion_analysis_and_object_tracking.html

20 Experimentation with model parameters showed that the parameter of a minimal accepted quality of image corners is the most sensitive (affects results the most).

S1.2 The Dense group

Distance inverse search (DIS) optical flow algorithm parameters:

- Finest level of the Gaussian pyramid on which the flow is computed (zero level corresponds to the original image resolution). The final flow is obtained by bilinear upscaling (default: 2; ranges: [0, 2^n])
- 25 – Maximum number of gradient descent iterations in the patch inverse search stage. Higher values may improve quality in some cases (default: 16; ranges: [1, N])
- Size of an image patch for matching (in pixels). Normally, default 8x8 patches work well enough in most cases (default: 8; ranges: [2, 2^n])
- Stride between neighbor patches. Must be less than patch size. Lower values correspond to higher flow quality (default: 4; ranges: [2, 2^n])
- 30 – Whether to use mean-normalization of patches when computing patch distance. It is turned on by default as it typically provides a noticeable quality boost because of increased robustness to illumination variations. Turn it off if you are certain that your sequence doesn't contain any changes in illumination (default: True; ranges: [False, True])

- Whether to use spatial propagation of good optical flow vectors. This option is turned on by default, as it tends to work better on average and can sometimes help recover from major errors introduced by the coarse-to-fine scheme employed by the DIS optical flow algorithm. Turning this option off can make the output flow field a bit smoother, however (default: True; ranges: [False, True])
- 5 – Weight of the smoothness term (default: 20; ranges: [0, 100])
- Weight of the color constancy term (default: 5; ranges: [1, 10])
- Weight of the gradient constancy term (default: 10; ranges: [1, 100])
- Number of fixed point iterations of variational refinement per scale. Set to zero to disable variational refinement completely. Higher values will typically result in more smooth and high-quality flow (default: 5; ranges: [0, 10])
- 10 Parameters values were set up based on default values proposed in the DIS implementation in the OpenCV library:

– https://docs.opencv.org/3.4/da/d06/classcv_1_1optflow_1_1DISOpticalFlow.html

PCAFflow global optical flow algorithm parameters:

- Number of basis vectors (default: (18, 14); ranges: [(1, 1), (N, N)])
- Sparse rate which controls density of sparse matches (default: 0.024; ranges: [0, 1])
- 15 – Retained corners fraction (default: 0.2; ranges: [0, 1])
- Occlusion threshold (default: 0.0003; ranges: [0, 1])
- Regularization term for solving least-squares. It is not related to the prior regularization (default: 0.00002; ranges: [0, 1])
- Clip parameter for Contrast Limited Adaptive Histogram Equalization (CLAHE) (default: 14; ranges: [1, N])

Parameters values were set up based on default values proposed in the PCAFlow implementation in the OpenCV library:

- 20 – https://docs.opencv.org/3.4/d1/da2/classcv_1_1optflow_1_1OpticalFlowPCAFflow.html

DeepFlow global optical flow algorithm parameters:

- Weight of the smoothness term (default: 1; ranges: [0.1, 10])
- Weight of the color constancy term (default: 0.5; ranges: [0.1, 1])
- Weight of the gradient constancy term (default: 5; ranges: [1, 100])
- 25 – Gaussian smoothing parameter (default: 0.6; ranges: [0.1, 1])
- Minimal dimension of an image in the pyramid (next, smaller images in the pyramid are generated until one of the dimensions reaches this size) (default: 25; ranges: [1, N])
- Scaling factor in the image pyramid (must be < 1) (default: 0.95; ranges: [0.1, 0.99])
- Number of iterations on each level of the pyramid (default: 5; ranges: [2, 10])
- 30 – Iterations of Succesive Over-Relaxation (SOR) (default: 25; ranges: [1, 100])
- Relaxation factor in SOR (default: 1.6; ranges: [1, 2])

Parameters values were set up based on default values proposed in the DeepFlow implementation in the OpenCV library:

- https://docs.opencv.org/3.4/d2/d84/group__optflow.html#ga5e735547610172695cb0b39495af9d6d
- https://github.com/opencv/opencv_contrib/blob/master/modules/optflow/src/deepflow.cpp

Farneback global optical flow algorithm parameters:

- Image scale to build pyramids (default: 0.5; ranges: [0.1, 0.9])
- 5
- Number of pyramid layers (default: 3; ranges: [1, 7])
 - Averaging window size (default: 15; ranges: [5, 30])
 - Number of iterations at each pyramid level (default: 3; ranges: [2, 10])
 - Size of the pixel neighborhood (default: 5; ranges: [3, 10])
 - Standard deviation of the Gaussian for smoothing derivatives (default: 1.1; ranges: [0.9, 2])
- 10
- Parameters values were set up based on trial-and-error experimentation and the following online source:
- http://docs.opencv.org/2.4/modules/video/doc/motion_analysis_and_object_tracking.html#calcopticalflowfarneback
- Experimentation with model parameters showed that the parameter of an averaging window size is the most sensitive (affects results the most).

Advection scheme properties:

- 15
- Type (default: constant-vector (the Dense model) or semi-lagrangian (the DenseRotation model))
 - Direction (default: backward; ranges: [forward, backward])

Interpolation algorithm:

- Type (default: idw (inverse distance weighting interpolatio from ω radlib.ipol), ranges: [idw, nearest, linear])

S2 Implementation of a warping for an advection of a precipitation field

Figure S1 shows the simplified example of warping implementation to the problem of precipitation field advection. We suppose that we have precipitation field with uniform intensity and sharp boundaries. This assumption allows us to perfectly identify source corners (or "features"). The second assumption is that each "feature" has the same displacement (vector field is collinear). This assumption allows us to avoid uncertainty related to corner tracking (we assume the perfect tracking), and we can precisely identify target corners. Based on these assumptions, we calculate the affine transformation matrix and use it to advect (warp, nowcast) the precipitation field.

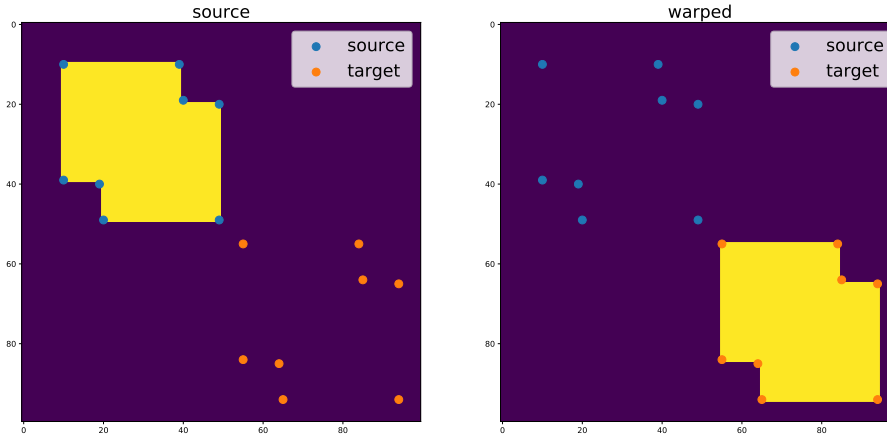


Figure S1. Result of well-defined warping operation

S3 Video clips of the selected events

You can find a Youtube playlist of the animated events by the following link:

10 <https://www.youtube.com/playlist?list=PLESStAZ1xRYDkzpj7YLo8Eo12VLNEIJw>

S4 Results in different categorical scores

We use the following categorical scores, where *hits*, *false alarms*, *misses*, and *correct negatives* are defined by the contingency table (Table S1) and the corresponding threshold value:

1. the false alarm ratio (FAR) – the fraction of falsely forecasted rain pixels to all forecasted rain pixels. FAR varies from 0 (best) to 1 (worst);

$$FAR = \frac{false\ alarms}{hits + false\ alarms}$$

2. the probability of detection (POD) – the fraction of correctly forecasted rain pixels to all rain pixels. POD varies from 0 (worst) to 1 (best);

$$POD = \frac{hits}{hits + misses}$$

3. the critical success index (CSI) – the fraction of correctly forecasted rain pixels to the sum of all rain pixels and false alarms. CSI varies from 0 (worst) to 1 (best);

$$CSI = \frac{hits}{hits + false\ alarms + misses}$$

4. the equitable threat score (ETS) – the adjusted and unbiased version of CSI, adjusted for the hits obtained by random chance. ETS varies from 1/3 (worst) and 1 (best) with 0 being the value of the random forecast.

$$ETS = \frac{hits - Dr}{hits + false\ alarms + misses - Dr}$$

$$Dr = \frac{(hits + false\ alarms)(hits + misses)}{hits + false\ alarms + misses + correct\ negatives}$$

Table S1. Contingency table for the categorical scores

Nowcast	Observation	
	Yes	No
Yes	hit	false alarm
No	miss	correct negative

Below you can find figures with FAR, POD, CSI, and ETS metrics for the selected events and different precipitation intensity threshold as follows:

- threshold = 0.125 mm h⁻¹ : Figure S2
- threshold = 0.25 mm h⁻¹ : Figure S3
- 5 – threshold = 0.5 mm h⁻¹ : Figure S4
- threshold = 1 mm h⁻¹ : Figure S5
- threshold = 5 mm h⁻¹ : Figure S6

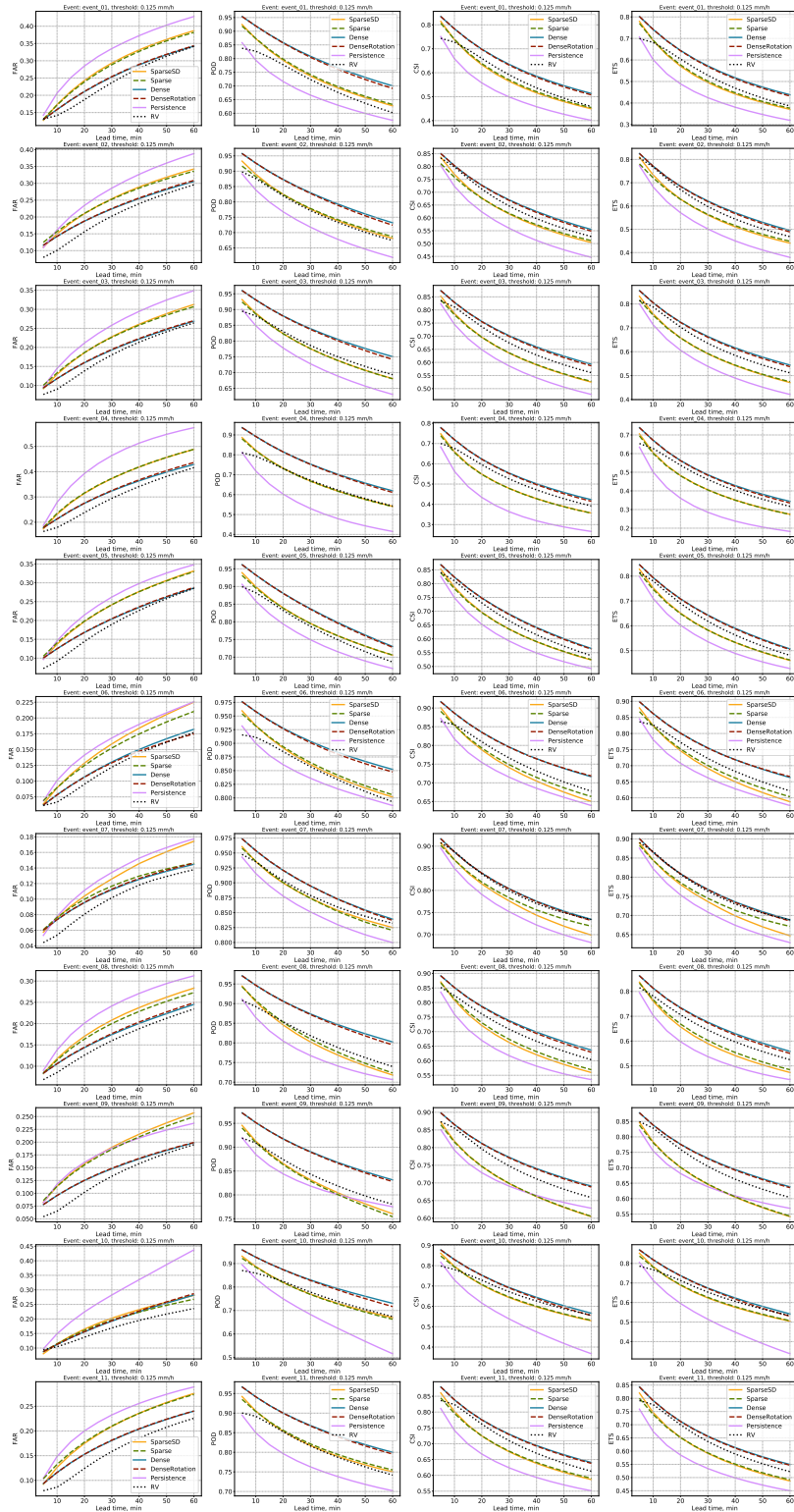


Figure S2. FAR, POD, CSI, and ETS for a rainfall threshold of 0.125 mm h⁻¹

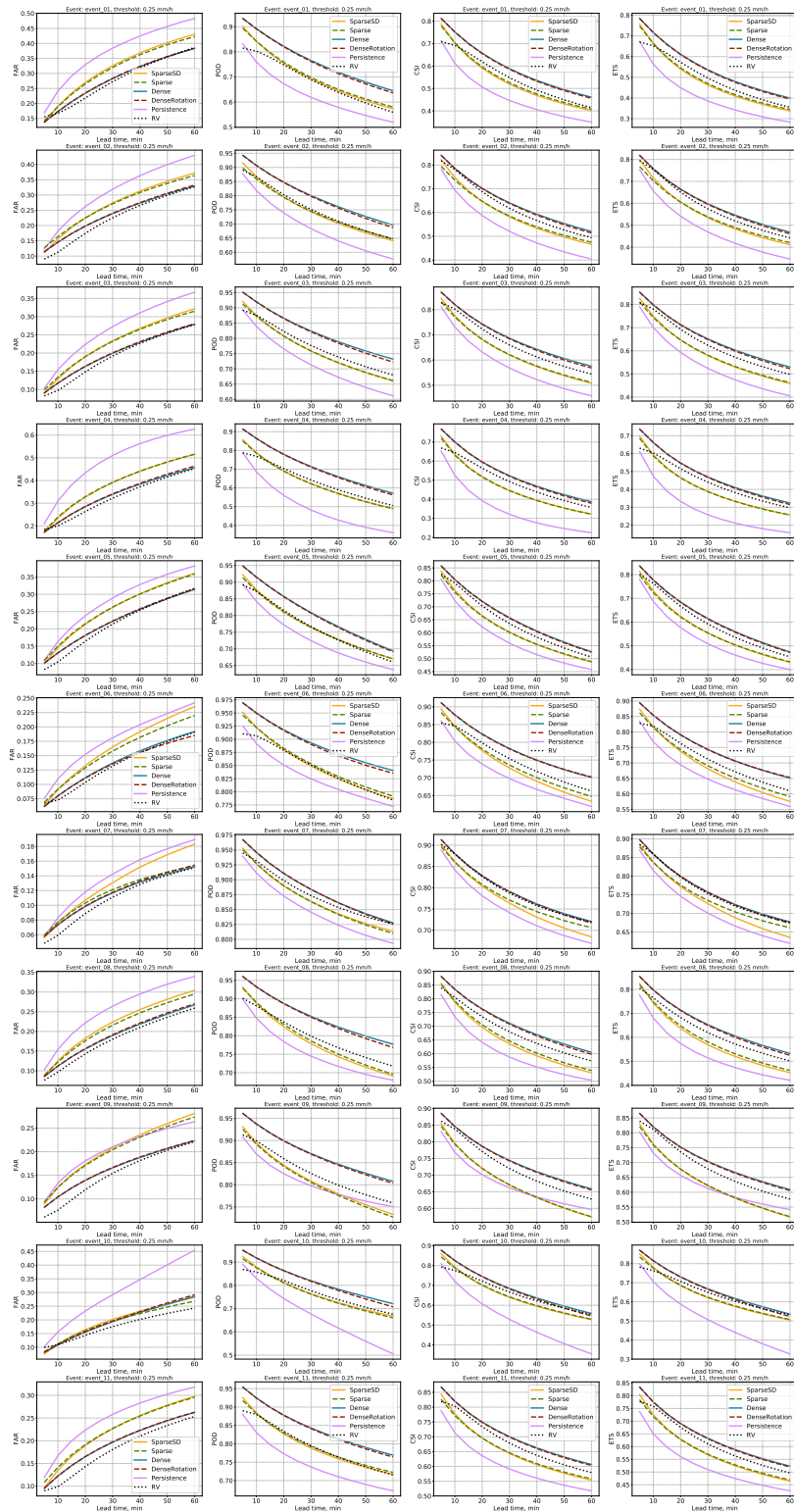


Figure S3. FAR, POD, CSI, and ETS for a rainfall threshold of 0.25 mm h^{-1}

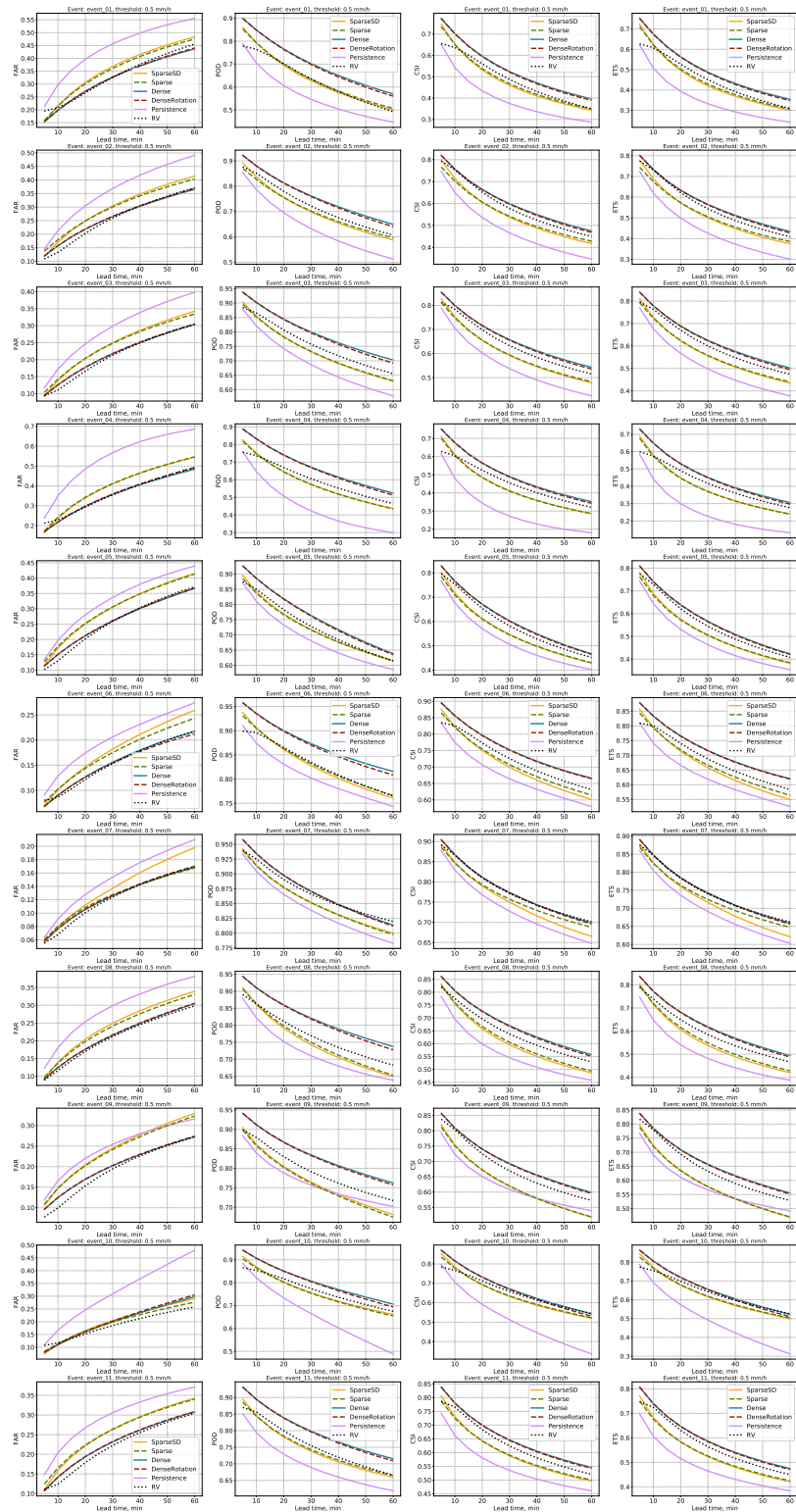


Figure S4. FAR, POD, CSI, and ETS for a rainfall threshold of 0.5 mm h^{-1}

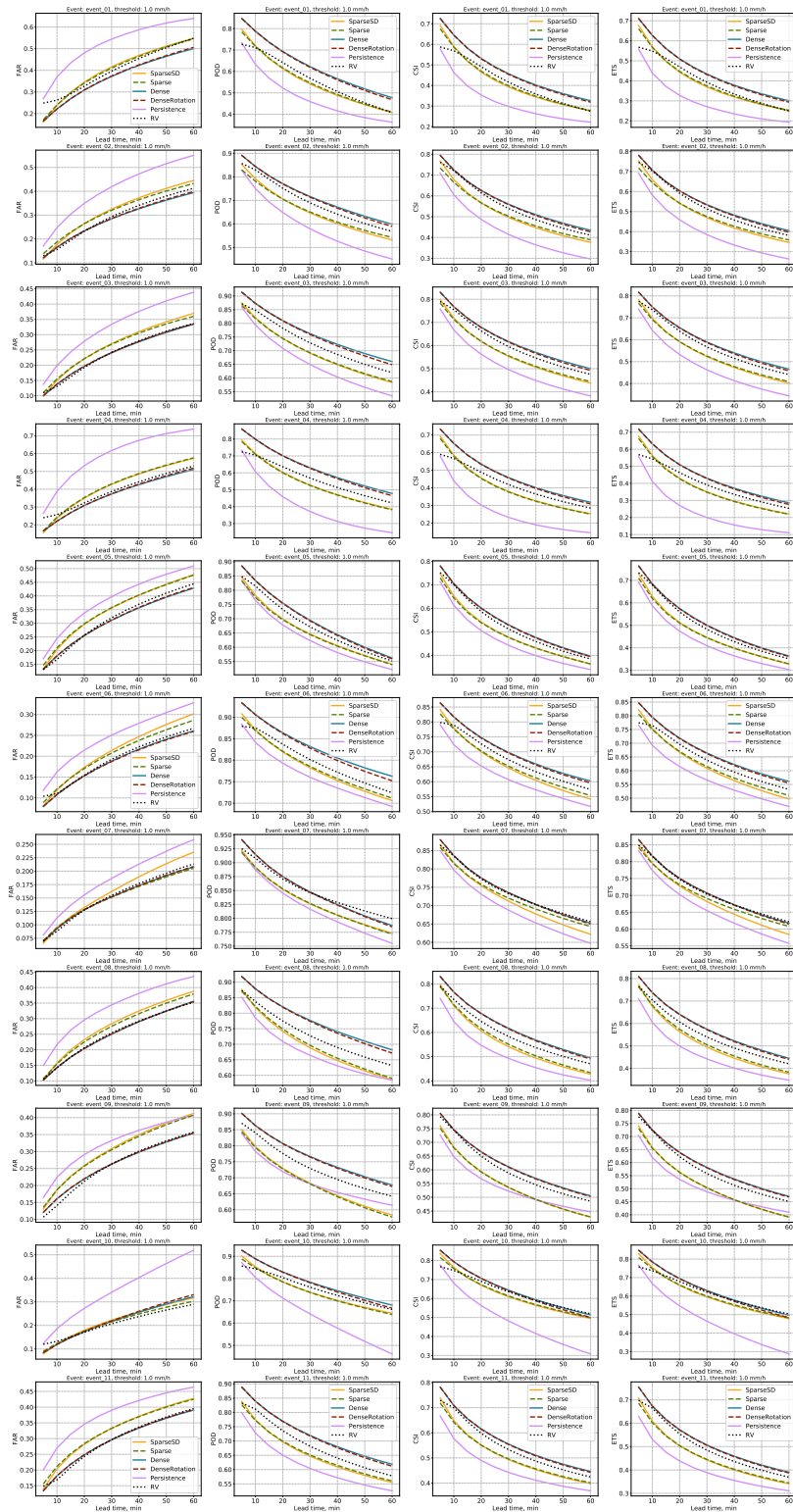


Figure S5. FAR, POD, CSI, and ETS for a rainfall threshold of 1 mm h⁻¹

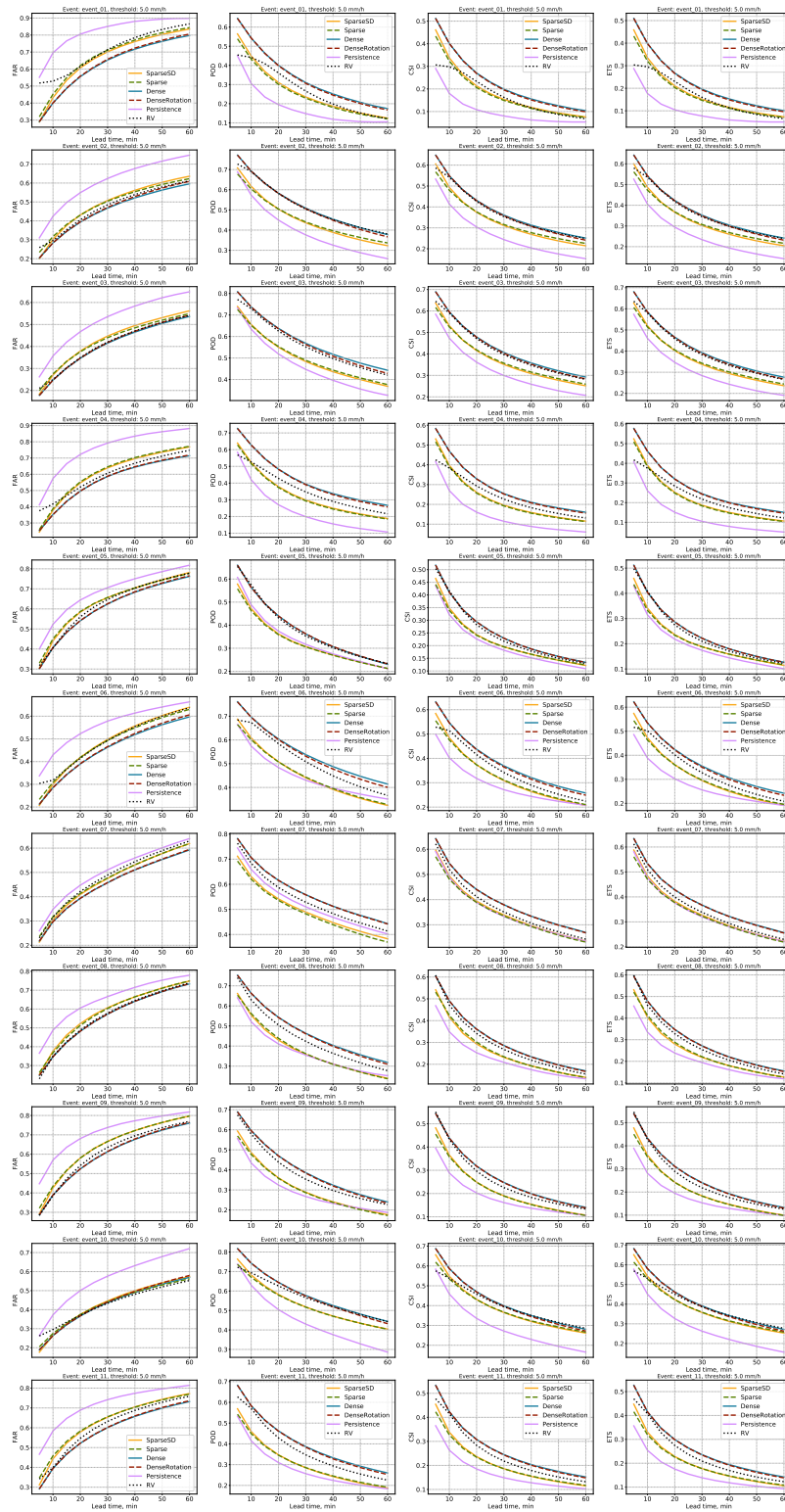


Figure S6. FAR, POD, CSI, and ETS for a rainfall threshold of 5 mm h^{-1}

S5 Intercomparison of different optical flow techniques

To compare the performance of different optical flow techniques available in the rainymotion library, we ran the DenseRotation model with the backward option of the semi-Lagrangian advection scheme under the same verification setting (Figures S7–S9). Results show that the DIS optical flow technique outperforms the remaining optical flow techniques while also providing the fastest computation (Table S2).

Table S2. Computational time of different optical flow techniques (mean \pm standard deviation of 100 runs)

Optical flow technique	Computational time
DIS	19.8 ms \pm 1.76 ms
PCAFlow	287 ms \pm 8.67 ms
Farneback	715 ms \pm 11.9 ms
DeepFlow	3.77 s \pm 117 ms

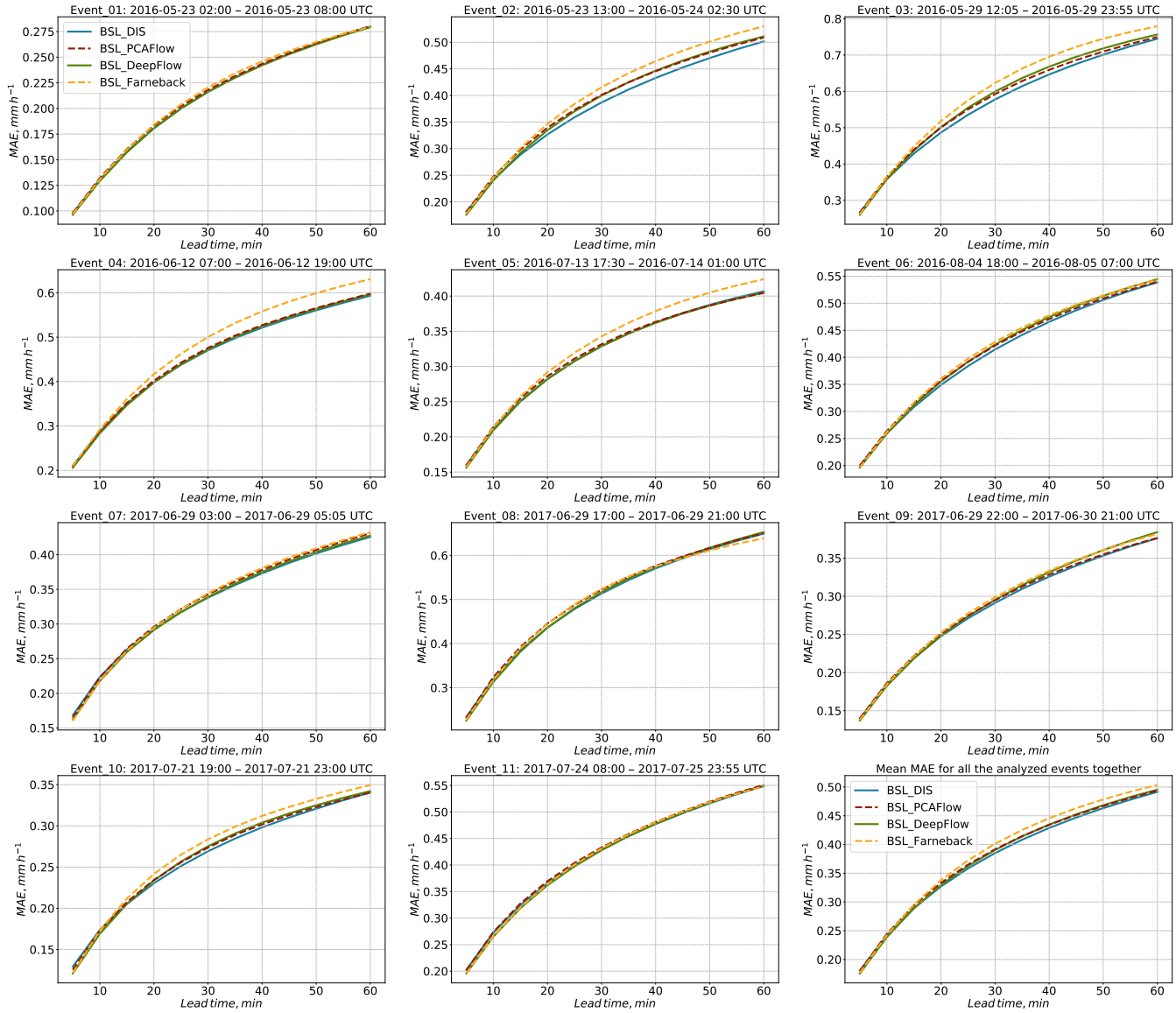


Figure S7. Lead time wise MAE for the proposed benchmark of different optical flow techniques

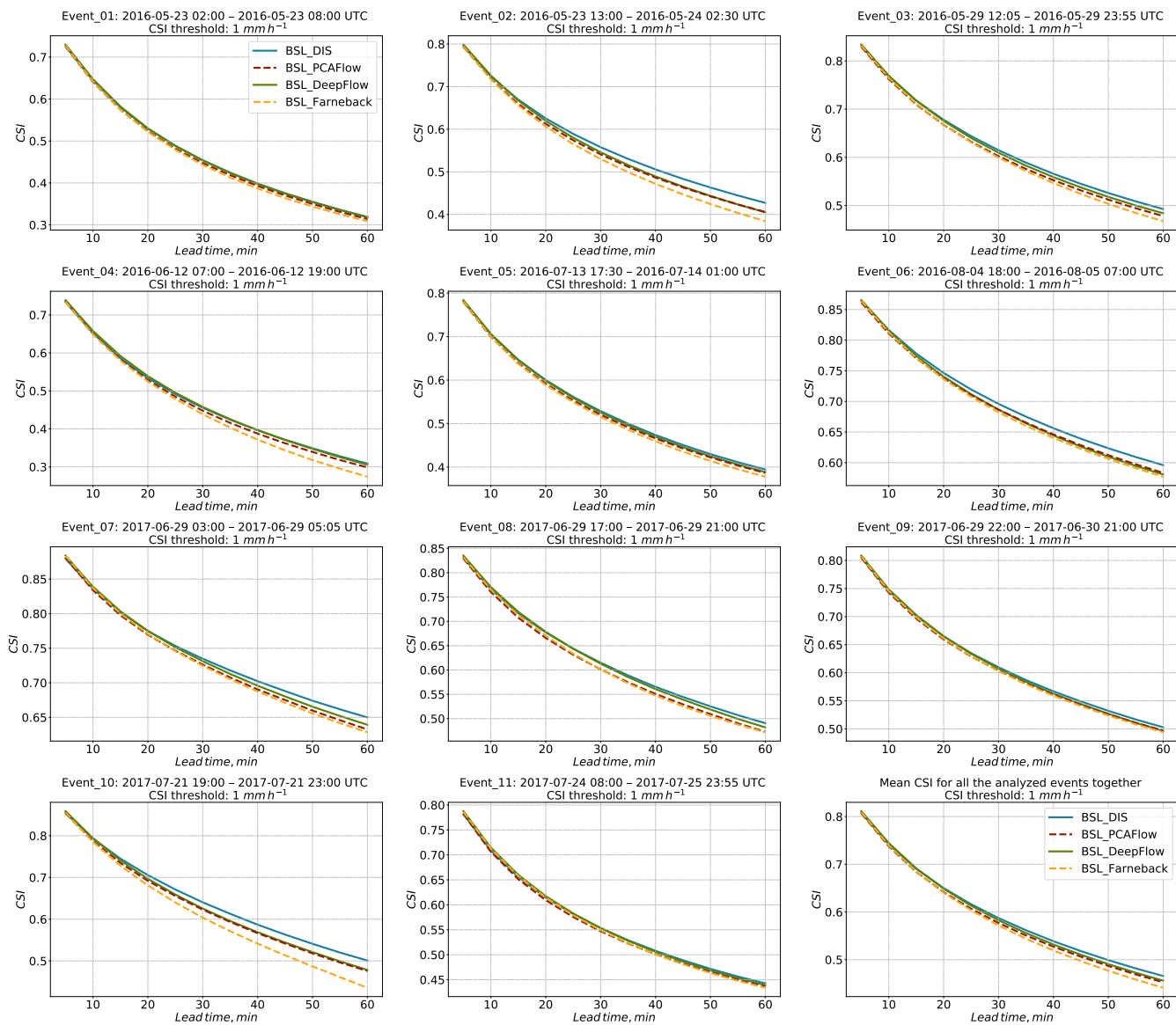


Figure S8. CSI for a rainfall threshold of 1 mm h^{-1} for the proposed benchmark of different optical flow techniques

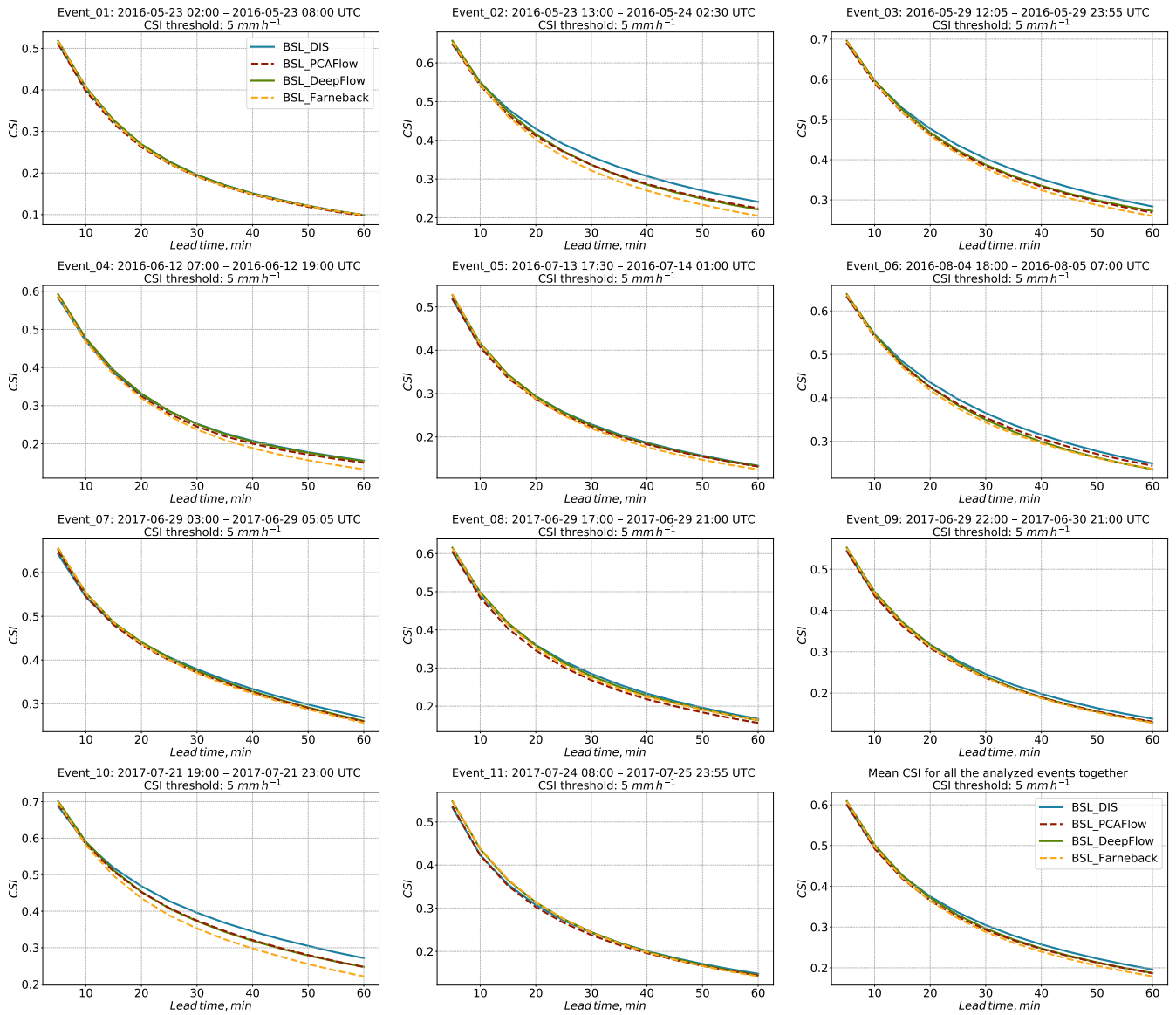


Figure S9. CSI for a rainfall threshold of 5 mm h^{-1} for the proposed benchmark of different optical flow techniques

S6 Intercomparison of different advection schemes

To compare the performance of different options in terms of displacement direction (i.e., forward and backward) of the constant-vector and semi-Lagrangian advection schemes, we ran both the Dense and DenseRotation models concerted with the utilization of the DIS optical flow algorithm under the same verification setting (Figures S10–S12). Results show that the backward scheme slightly outperforms the forward one.

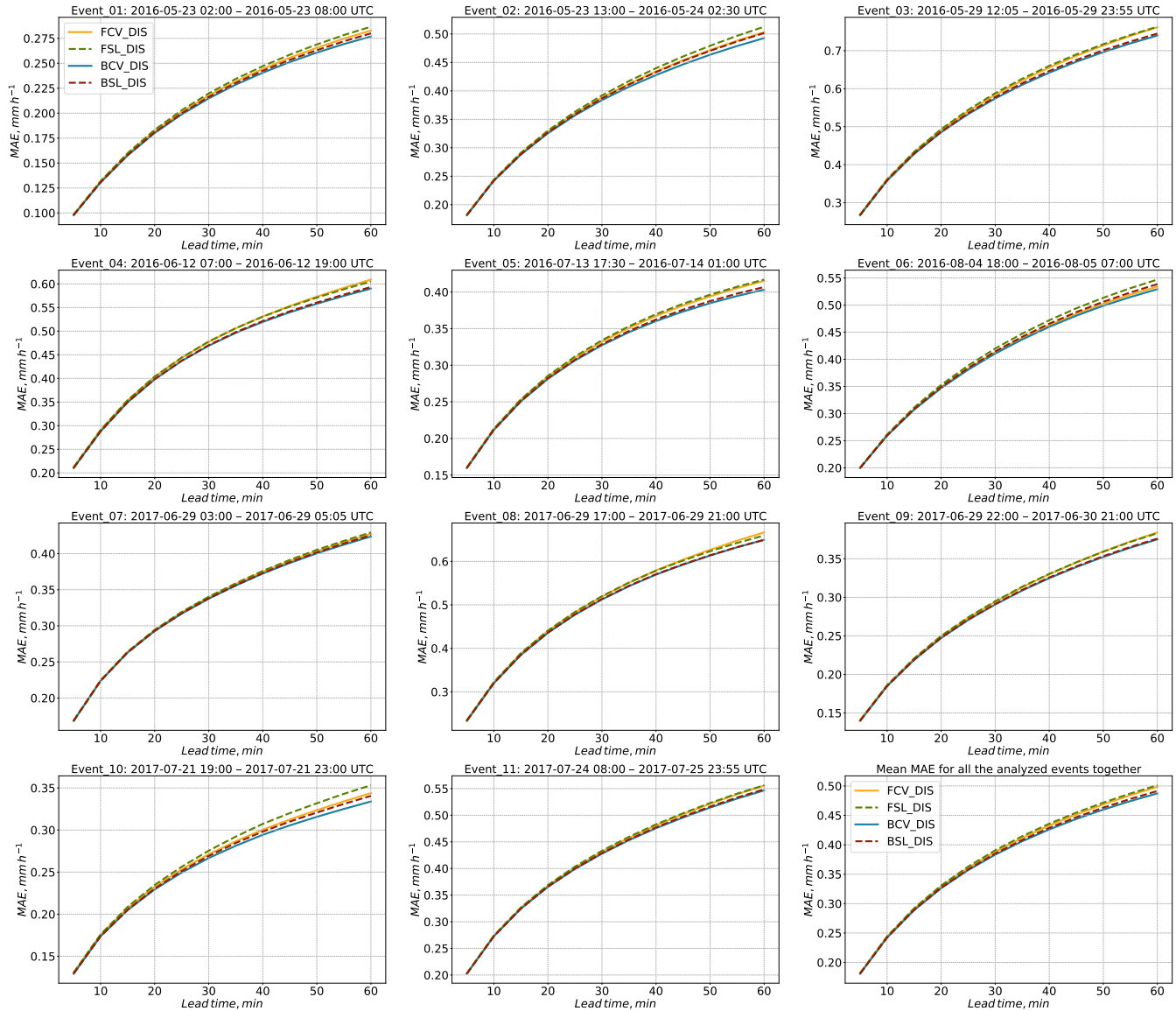


Figure S10. Lead time wise MAE for the proposed benchmark of different advection schemes

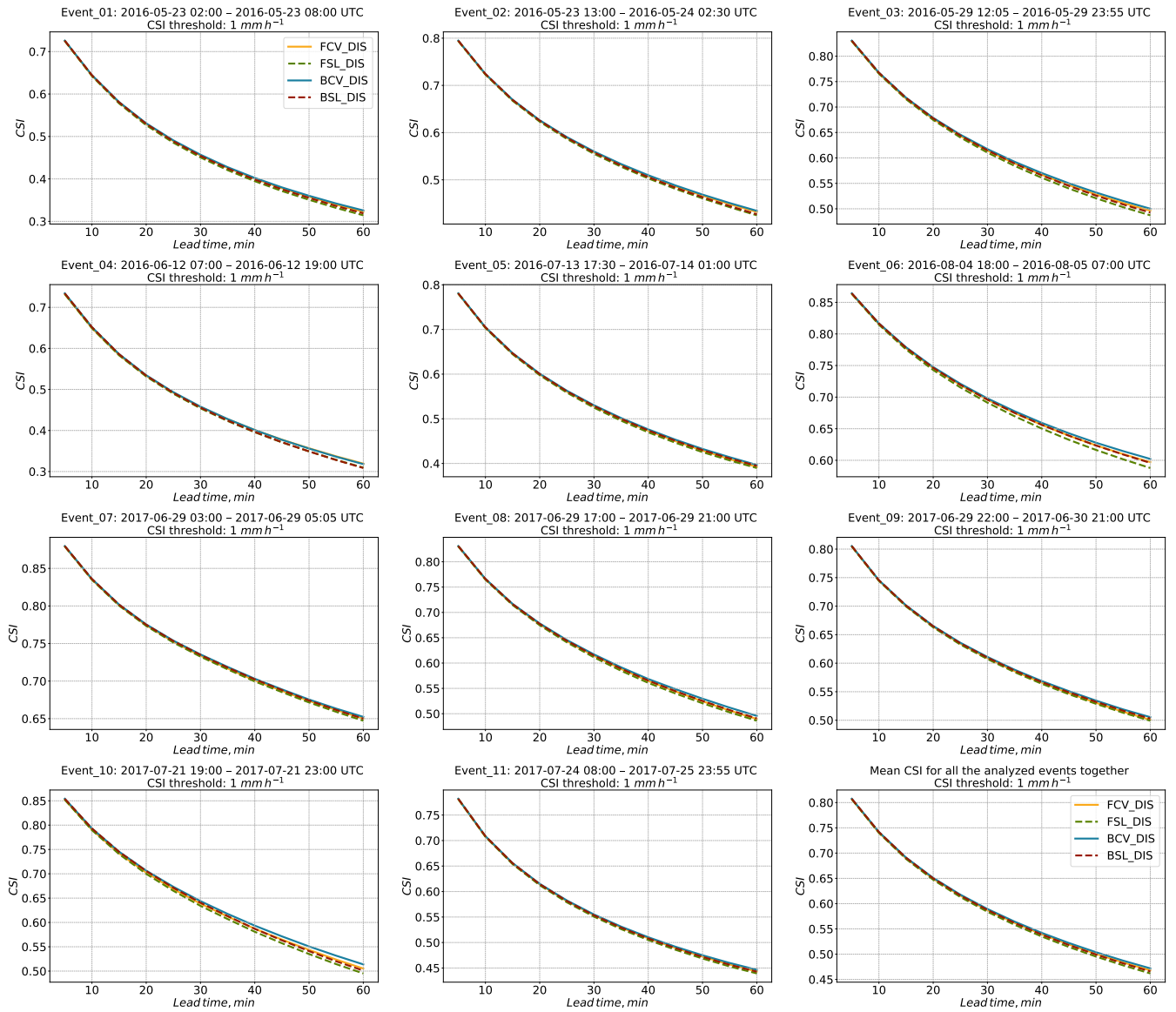


Figure S11. CSI for a rainfall threshold of 1 mm h^{-1} for the proposed benchmark of different advection schemes

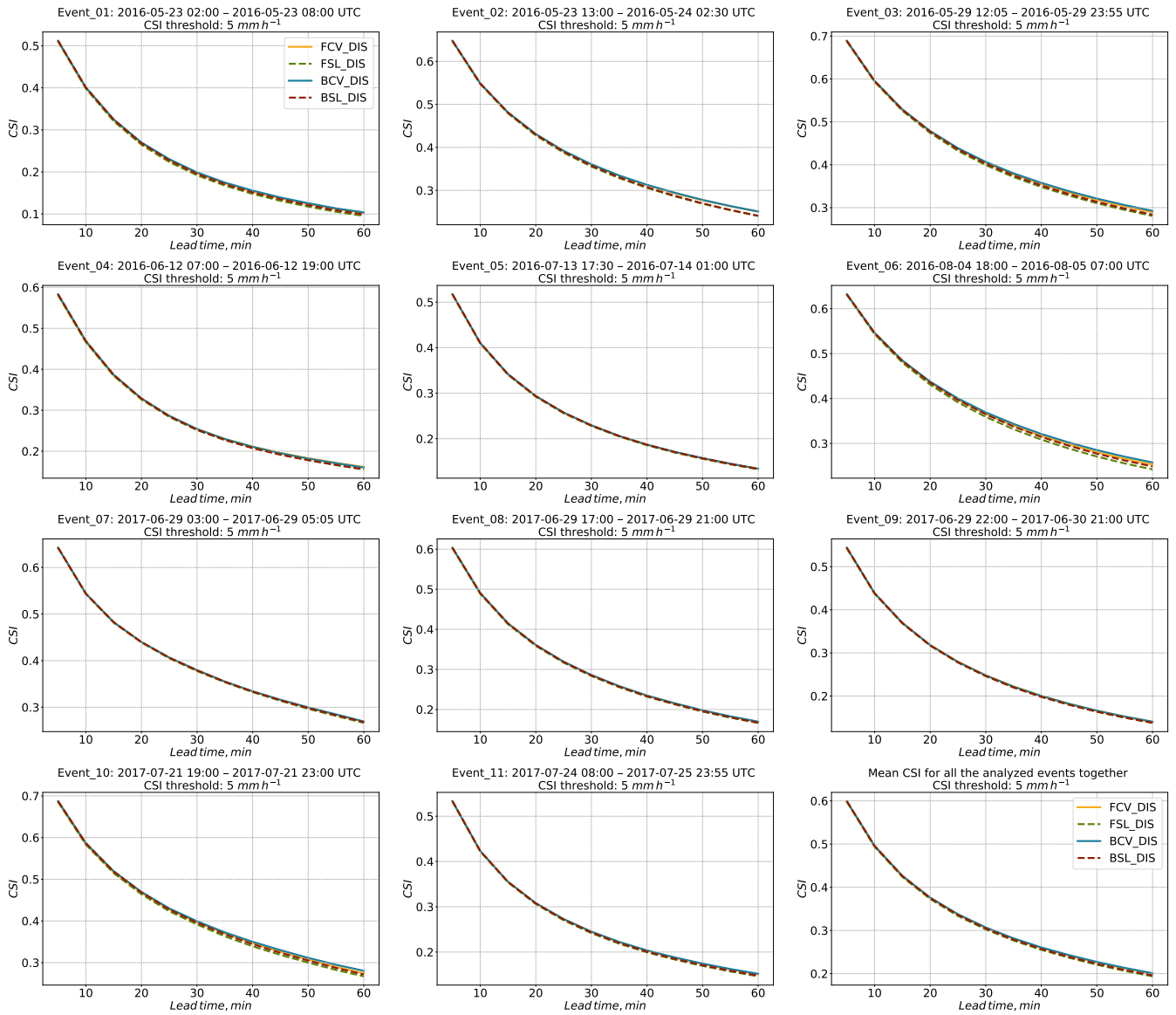


Figure S12. CSI for a rainfall threshold of 5 mm h^{-1} for the proposed benchmark of different advection schemes