Final response in the interactive discussion

Dear Referees, dear Editor,

We would like to thank you very much for your positive comments and constructive suggestions to our manuscript *"RainNet v1.0: a convolutional neural network for radar-based precipitation nowcasting"*.

In this document, we would like to provide our responses to the comments of each of the referees in one single document and to outline the corresponding changes to the manuscript. We will represent the referee comments in **bold** font, and our responses in normal font. Quotations from the original manuscript will be in *italics*, changes as part of the manuscript revision will be highlighted as underlined. For the sake of clarity and brevity, we have omitted the introductory parts of the referee reports (this omittance is marked as [...]).

We hope that our response together with the revision of the manuscript sufficiently addresses the referees' concerns.

Sincerely,
Georgy Ayzel (on behalf of the author team)

Referee comment #1 (by Gabriele Franch)

[...] I recommend this study for publication after considering the minor comments listed below:

1. **In ll. 167-172, can you provide the exact number of the train (optimization), validation and test (verification) sequences? Moreover, can you explain how the sequences are extracted from the dataset (are the sequences extracted using an overlapping rolling window over the selected time periods?)**

We thank the referee for the suggestion: the required details should, in fact, be provided to the reader. We will add, in the revised manuscript, the corresponding information (i.e. the number of sequences and whether the sequences overlap) at the end of the paragraph that was mentioned by the referee - as follows (ll. 172-175 of the revised manuscript):

In this study, we use RY data that covers the period from 2006 to 2017. We split the available RY data as follows: while we use data from 2006 to 2013 to optimize RainNet's model parameters and data from 2014 to 2015 to validate RainNet performance, data from 2016 to 2017 is used for model verification (Sect. 3.3). For both optimization and validation

periods, we keep only data from May to September and ignore time steps for which the precipitation field (with rainfall intensity more than 0.125 mm h−1) covers less than 10% of the RY domain. For each subset of the data - for optimization, validation, and verification -, every time step (or frame) is used once as t_0 (forecast time) so that the resulting sequences that are used as input to a single forecast (t_0-15 min, …, t_0) overlap in time. The number of resulting sequences amounts to 41988 for the optimization, 5722 for the validation, and 9626 for the verification (see also Sect. 3.3).

2. **Given the analysis of the power spectrum and the reported smoothing in the prediction, it seems that RainNet may suffer from a severe underestimation of high rainfall rates. In this regard, it would be extremely beneficial to include a higher rain rate than 5mm/h for the analysis of the categorical scores. This would also help provide a better comparison with Rainymotion and can help to answer Line 230 that states: "RainNet might have difficulties in predicting pronounced precipitation features with high intensities". Thus, I suggest adding also at least one heavy rainfall threshold (FSS and CSI≥15mm/h) in the analysis.**

We agree that a threshold of 5 mm/h does not qualify as heavy rainfall, although the results for a threshold of 5 mm/h already provide a good impression about the general effects of an increasing intensity threshold for a categorical metric such as the CSI: first, a strong loss of skill over *all* lead times and *all* competing methods, and, second, a relatively stronger loss of skill for RainNet in comparison to rainymotion.

For intensity thresholds of 10 and 15 mm/h, these effects become much more prominent, so that rainymotion, in fact, outperforms RainNet (in terms of CSI) - particularly at intermediate lead times between from 10 to 50 minutes. At the same time, the CSI of both methods becomes so low that neither can be really considered as skillful in predicting the exceedance of higher thresholds.

Hence, we would like to thank the referee very much for his suggestion. Looking at thresholds of 10 and 15 mm/h is in fact revealing: while the results basically confirm our assumptions that were based on the 5 mm/h threshold, they demonstrate more clearly how difficult it is for RainNet to learn the prediction of high intensity features. Whether this is an effect of the spatial smoothing, or whether RainNet has specifically learned, based on the current training setting and loss function, that it is efficient to “attenuate” high rainfall intensities, will be subject to future research.

As a consequence, we will include these additional results in the main manuscript: Fig. 3 will be extended by a CSI for 10 and 15 mm/h, and Fig. 6 will be extended by the FSS values that correspond to threshold intensities of 10 and 15 mm/h (please see the new versions of these figures below). The main text in Section 4 (Results and discussion) and Section 5 (Summary and conclusions) will be changed accordingly in the revised version of the manuscript in order to explicitly account for these additional results.

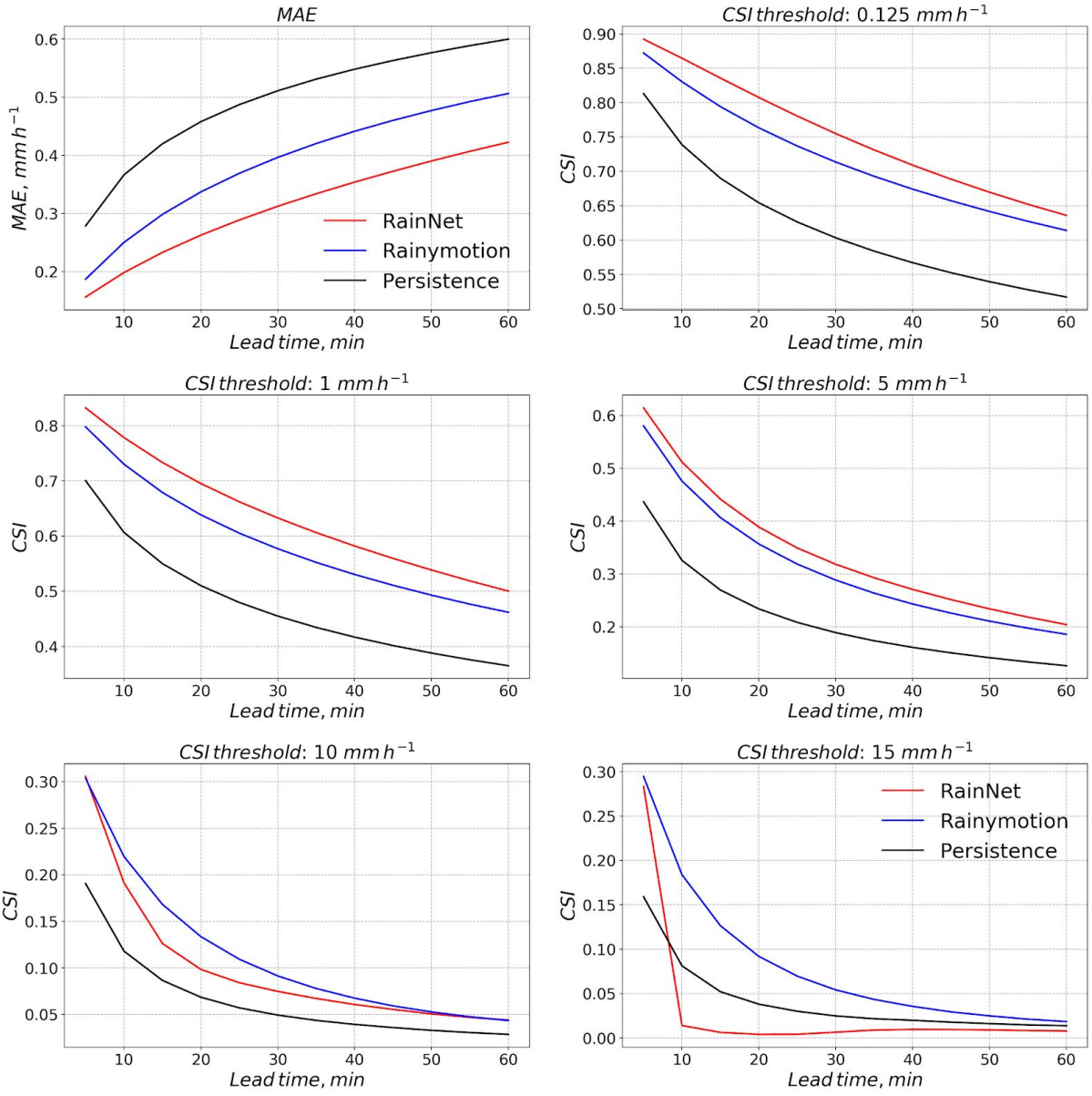

Figure 3 (revised). Mean Absolute Error (MAE) and Critical Success Index (CSI) for three different intensity thresholds (0.125 mm h^{-1}, 1 mm h^{-1}, 5 mm h^{-1}, 10 mm h^{-1}, 15 mm h^{-1}). The metrics are shown as a function of lead time. All values represent the average of the corresponding metric over all 11 verification events.

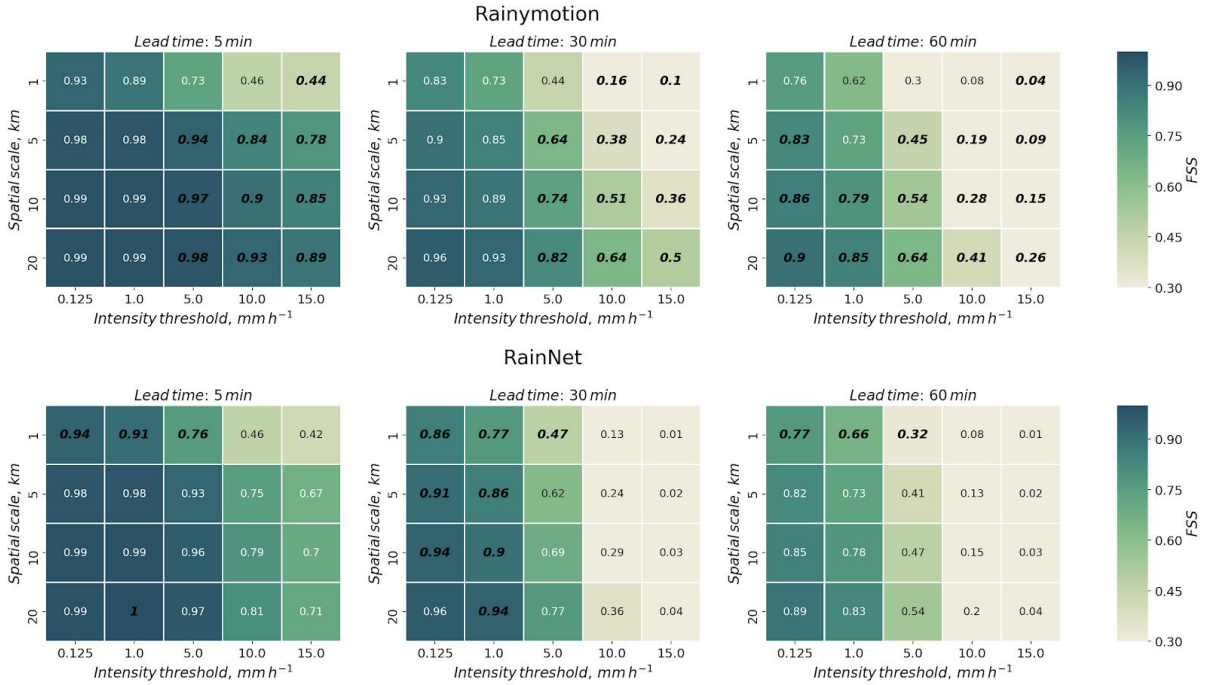

Figure 6 (revised). Fractions Skill Score (FSS) for Rainymotion (top panel) and RainNet (bottom panel), for 5, 30, and 60 minutes lead time, and spatial window sizes of 1, 5, 10 and 20 km, and for intensity thresholds of 0.125, 1, 5, 10 and 15 mm h^{-1}. In addition to the color code of the FSS, we added the numerical FSS values. The FSS value of the model which is significantly superior for a specific combination of window size, intensity threshold, and lead time is typed in bold black digits, for the inferior model in regular.

Referee comment #2 (by Scott Collis)

[...] I only have four minor suggestions (and they are just that, suggestions):

1. **The training and prediction was entirely based on DWD RY gridded rainfall product. I think the authors should discuss the applicability of training a network on one data set and application to another. The CNN learns what features propagate and dissipate (an advantage over purely advective techniques) but this may not apply in regions where different physics dominate.**

We fully agree with the referee. If the network was, in fact, successful in learning initiation, growth and dissipation of features, the transferability of such a trained network to a different region of application would most likely be limited. That limitation would be most pronounced in areas and situations in which precipitation dynamics are dominated by recurrent drivers and processes (such as features of the general circulation or orography). In ll. 351-359 of the original manuscript, however, we already pointed out that the current (trained) network is most likely very limited in its ability to represent precipitation dynamics. While we hope to change that in the future, it implies, in turn, that the transferability to another region/dataset might in fact not be as low as expected. Yet, there is only one way to find out: by actually

carrying out a verification experiment with RainNet on a dataset from another region, using the pretrained weights... another item on the to-do list for future research.

We would like to avoid to put too much emphasis on this discussion in the manuscript (as it is quite speculative), but we will account for the referee's suggestion by adding the following sentence to the paragraph in ll. 376-377 of the revised manuscript:

[...] implies that RainNet, in essence, learned to represent motion patterns and optimal smoothing. In that case, the trained model might even be applicable on data in another region which could be tested in future verification experiments.

2. **It would be good for the authors to discuss a little more on what would be required of the input data. Can a potential user train with any NDArray style data?**

In principle, the input data is of class numpy.ndarray (float32 with no missing values). However, the dataset used for training is so large that it cannot be read into memory at once. We store data directly in `.npy` files as it is more efficient in terms of parallelization and utilization of computational resources. While we are happy to provide this response to the referee, we would prefer not to go into these technical details in the manuscript.

3. **On line 155 where training times are discussed it would be good, for the understanding of readers to restate how many frames (radar time steps) were used in the training. This would be a repetition but I believe it would add to the readers understanding.**

We assume that the referee refers to the number of time steps or frames used as training data, not the number of frames that are used as model input for a single prediction (which is the four recent frames, see l. 116 of the revised manuscript). Stating that information, as required by the referee, would not be repetition as it has not been stated in the original manuscript. In fact, the referee's request is well in line with the first issue raised by referee #1 (please see above). We would prefer, however, to provide the required details in ll. 172-175 of the revised manuscript (not, as suggested by the referee, around l. 155 of the original manuscript).

4. **In the author's section on future research I am surprised not to see other atmospheric data inputs/layers talked about. If I understood the paper correctly the CNN is trained purely on image-like data with no environmental awareness. I wonder if the evolution (again information that can not be deduced by simple advection) could be better predicted with information like precipitable water or information about terrain? The developing area of physics aware machine learning could be an area to explore.**

We would like to refer to ll. 360-366 of the original manuscript. In that paragraph, we already outline the perspective to use output fields of atmospheric models or 3-d polarimetric radar moments as input layers that are more physically meaningful. We hope that pointing out these perspectives is sufficient to meet the referee's requirements.

RainNet v1.0: a convolutional neural network for radar-based precipitation nowcasting

Georgy Ayzel[1], Tobias Scheffer[2], and Maik Heistermann[1]

[1]Institute for Environmental Sciences and Geography, University of Potsdam, Potsdam, Germany
[2]Department of Computer Science, University of Potsdam, Potsdam, Germany

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

Abstract. In this study, we present RainNet, a deep convolutional neural network for radar-based precipitation nowcasting. Its design was inspired by the U-Net and SegNet families of deep learning models which were originally designed for binary segmentation tasks. RainNet was trained to predict continuous precipitation intensities at a lead time of five minutes, using several years of quality-controlled weather radar composites provided by the German Weather Service (DWD). That data set covers Germany with a spatial domain of 900×900 km, and has a resolution of 1 km in space and 5 minutes in time. Independent verification experiments were carried out on eleven summer precipitation events from 2016 to 2017. In order to achieve a lead time of one hour, a recursive approach was implemented by using RainNet predictions at five minutes lead time as model input for longer lead times. In the verification experiments, trivial Eulerian persistence and a conventional model based on optical flow served as benchmarks. The latter is available in the *rainymotion* library, and had previously been shown to outperform DWD's operational nowcasting model for the same set of verification events.

RainNet significantly outperforms the benchmark models at all lead times up to 60 minutes for the routine verification metrics Mean Absolute Error (MAE) and the Critical Success Index (CSI~~,~~) at intensity thresholds of 0.125, 1, and 5 mm h^{-1}~~).~~ ~~Apart from its superiority in terms of MAE and CSI, an undesirable property of RainNet predictions is , however, the~~ . However, Rainymotion turned out to be superior in predicting the exceedance of higher intensity thresholds (here 10 and 15 mm h^{-1}). The limited ability of RainNet to predict heavy rainfall intensities is an undesirable property which we attribute to a high level of spatial smoothing introduced by the model. At a lead time of five minutes, an analysis of Power Spectral Density confirmed a significant loss of spectral power at length scales of 16 km and below. Obviously, RainNet had learned an optimal level of smoothing to produce a nowcast at 5 minutes lead time. In that sense, the loss of spectral power at small scales is informative, too, as it reflects the limits of predictability as a function of spatial scale. Beyond the lead time of five minutes, however, the increasing level of smoothing is a mere artifact – an analogue to numerical diffusion – that is not a property of RainNet itself, but of its recursive application. In the context of early warning, the smoothing is particularly unfavourable since pronounced features of intense precipitation tend to get lost over longer lead times. Hence, we propose several options to address this issue in prospective research, including an adjustment of the loss function for model training, model training for longer lead times, and the prediction of threshold exceedance in terms of a binary segmentation task. Furthermore, we suggest additional input data that could help to better identify situations with imminent precipitation dynamics. The model code, pretrained weights, and training data are provided in open repositories as an input to such future studies.

1 Introduction

The term *nowcasting* refers to forecasts of precipitation field movement and evolution at high spatiotemporal resolution (1–10 minutes, 100–1000 meters) and short lead times (minutes to a few hours). Nowcasts have become popular not only to a broad civil community for planning everyday activities; they are particularly relevant as part of early warning systems for heavy rainfall, and related impacts such as flash floods or landslides. While the recent advances in high-performance computing and data assimilation significantly improved numerical weather prediction (NWP) (Bauer et al., 2015), the computational resources required to forecast precipitation field dynamics at very high spatial and temporal resolution are typically prohibitive for the frequent update cycles (5–10 minutes) that are required for operational nowcasting systems. Furthermore, the heuristic extrapolation of precipitation dynamics that are observed by weather radars still outperform NWP forecasts at short lead times (Lin et al., 2005; Sun et al., 2014). Thus, the development of new nowcasting systems based on parsimonious, but reliable and fast techniques, remains an essential trait in both atmospheric and natural hazards research.

There are many nowcasting systems which work operationally all around the world to provide precipitation nowcasts (Reyniers, 2008; Wilson et al., 1998). These systems, in their core, utilize a two-step procedure that was originally suggested by Austin and Bellon (1974), consisting of tracking and extrapolation. In the tracking step, a velocity is obtained from a series of consecutive radar images. In the extrapolation step, that velocity is used to propagate the most recent precipitation observation into the future. Various flavors and variations of this fundamental idea have been developed and operationalized over the past decades, and provide value to users of corresponding products. Still, the fundamental approach to nowcasting has not changed much over the recent years – a situation that might change with the increasing popularity of deep learning in various scientific disciplines.

Deep learning refers to machine-learning methods for artificial neural networks with "deep" architectures. Rather than relying on engineered features, deep learning derives low-level image features on the lowest layers of a hierarchical network, and increasingly abstract features on the high-level network layers, as part of the solution of an optimization problem based on training data (LeCun et al., 2015). Deep learning took its rise from the field of computer science when it started to dramatically outperform reference methods in image classification (Krizhevsky et al., 2012), machine translation (Sutskever et al., 2014), followed by speech recognition (LeCun et al., 2015). Three main reasons caused this substantial breakthrough in predictive efficacy: the availability of "big data" for model training, the development of activation functions and network architectures that result in numerically stable gradients across many network layers (Dahl et al., 2013), and the ability to scale the learning process massively by parallelization on graphics processing units (GPUs). Today, deep learning is rapidly spreading in many data-rich scientific disciplines, and complements researchers' toolboxes with efficient predictive models, including the field of geosciences (Reichstein et al., 2019).

But while expectations in atmospheric sciences are high (see e.g., Dueben and Bauer, 2018; Gentine et al., 2018), the investigation of deep learning in radar-based precipitation nowcasting is still in its infancy, and universal solutions are not yet available. Shi et al. (2015) were the first to introduce deep learning models in the field of radar-based precipitation nowcasting: they presented a Convolutional Long Short-Term Memory architecture (ConvLSTM) which outperformed the optical flow

based ROVER nowcasting system in the Hong Kong area. A follow-up study (Shi et al., 2017) introduced new deep learning architectures, namely the Trajectory Gated Recurrent Unit (TrajGRU) and the Convolutional Gated Recurrent Unit (ConvGRU), and demonstrated that these models outperform the ROVER nowcasting system, too. Further studies of Singh et al. (2017) and Shi et al. (2018) confirmed the potential of deep-learning models for radar-based precipitation nowcasting for different sites in the US and China. Most recently, Agrawal et al. (2019) introduced a U-net based deep learning model for the prediction of exceedance of specific rainfall intensity thresholds compared to optical flow and numerical weather prediction models. Hence, the exploration of deep learning techniques in radar-based nowcasting has begun, and the potential to overcome the limitations of standard tracking and extrapolation techniques has become apparent. There is a strong need, though, to further investigate different architectures, to set up new benchmark experiments, and to understand under which conditions deep learning models can be a viable option for operational services.

In this paper, we introduce RainNet – a deep neural network which aims at learning representations of spatiotemporal precipitation field movement and evolution from a massive open radar data archive to provide skillful precipitation nowcasts. The present study outlines RainNet's architecture and its training, and reports on a set of benchmark experiments in which RainNet competes against a conventional nowcasting model based on optical flow. Based on these experiments, we evaluate the potential of RainNet for nowcasting, but also its limitations in comparison to conventional radar-based nowcasting techniques. Based on this evaluation, we attempt to highlight options for future research towards the application of deep learning in the field of precipitation nowcasting.

2 Model description

2.1 Network architecture

To investigate the potential of deep neural networks for radar-based precipitation nowcasting, we developed RainNet – a convolutional deep neural network (Fig. 1). Its architecture was inspired by the U-Net and SegNet families of deep learning models for binary segmentation (Badrinarayanan et al., 2017; Ronneberger et al., 2015; Iglovikov and Shvets, 2018). These models follow an encoder-decoder architecture in which the encoder progressively downscales the spatial resolution using pooling, followed by convolutional layers; and the decoder progressively upscales the learned patterns to a higher spatial resolution using upsampling, followed by convolutional layers. There are skip connections (Srivastava et al., 2015) from the encoder to the decoder in order to ensure semantic connectivity between features on different layers.

As elementary building blocks, RainNet has 20 convolutional, four max pooling, four upsampling, two dropout layers, and four skip connections. Convolutional layers aim to generate data-driven spatial features from the corresponding input volume using several convolutional filters. Each filter is a 3D tensor of learnable weights with a small spatial kernel size (e.g., 3×3, and the third dimension equal to that of the input volume). A filter convolves through the input volume with a step size parameter (or stride, stride=1 in this study) and produces a dot product between filter weights and corresponding input volume values. A bias parameter is added to this dot product, and the results are transformed using an adequate activation function. The purpose of the activation function is to add nonlinearities to the convolutional layer output – to enrich it to learn non-linear features.

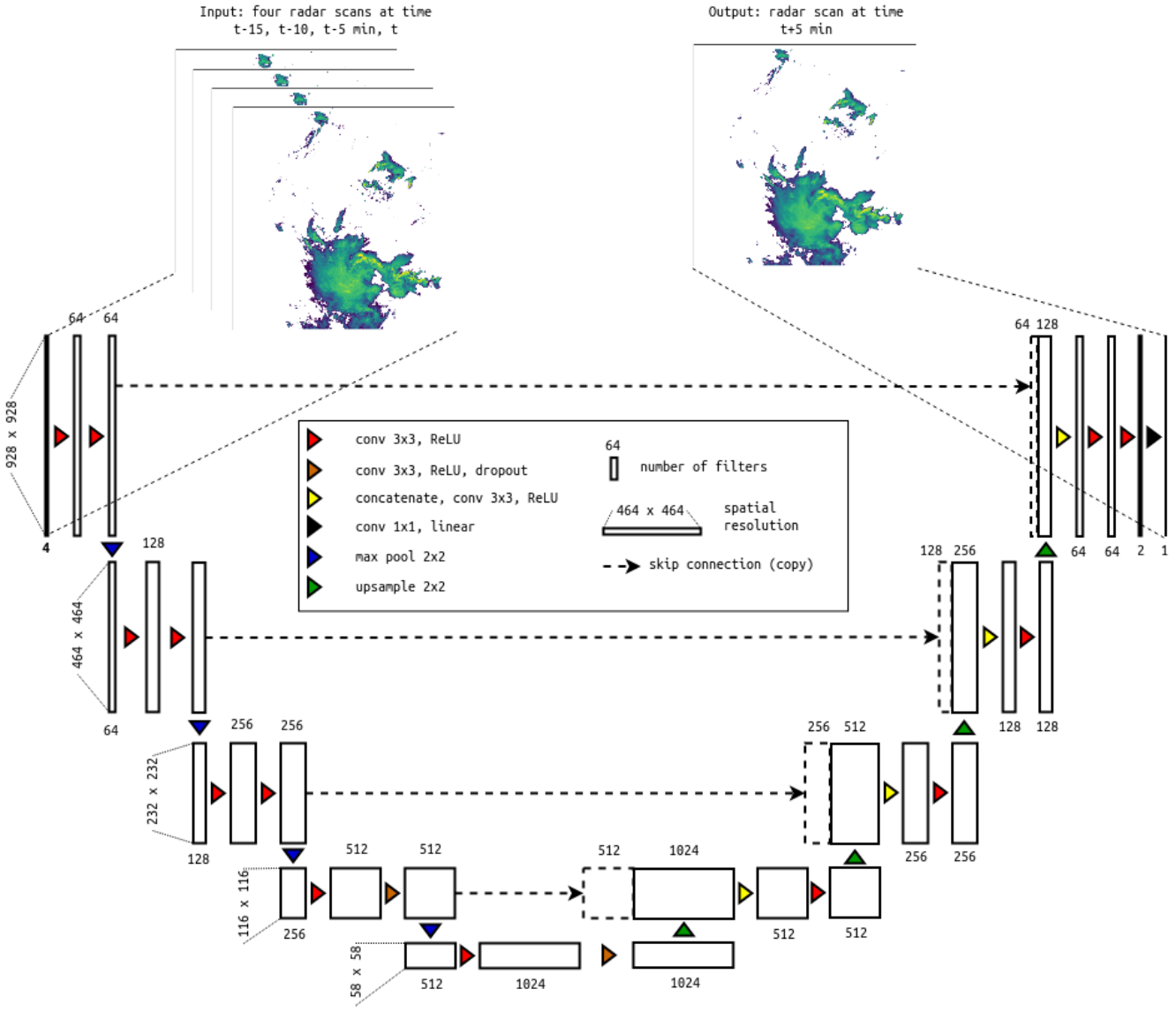

Figure 1. Illustration of the RainNet architecture. RainNet is a convolutional deep neural network which follows a standard encoder-decoder structure with skip connections between its branches. See main text for further explanation.

To increase the efficiency of convolutional layers, it is necessary to optimize their hyperparameters (such as number of filters, kernel size, and type of activation function). This has been done in a heuristic tuning procedure (not shown). As a result, we use convolutional layers with up to 1024 filters, kernel sizes of 1×1 and 3×3, and linear or Rectified Linear Unit (ReLU; Nair and Hinton, 2010) activation functions.

Using a max pooling layer has two primary reasons: it achieves an invariance to scale transformations of detected features, and increases the network's robustness to noise and clutter (Boureau et al., 2010). The filter of a max pooling layer slides over the input volume independently for every feature map with some step parameter (or stride) and resizes it spatially using the *maximum* (max) operator. In our study, each max pooling layer filter is of size 2×2, applied with a stride of 2. Thus, we take the maximum of 4 numbers in the filter region (2×2) which downsamples our input volume by factor 2. In contrast to a max pooling layer, an upsampling layer is designed for spatial upsampling of the input volume (Long et al., 2015). An upsampling layer operator slides over the input volume and fills (copies) each input value to a region that is defined by the upsampling kernel size (2×2 in this study).

Skip connections were proposed by Srivastava et al. (2015) in order to avoid the problem of vanishing gradients for the training of very deep neural networks. Today, skip connections are a standard group of methods for any form of information transfer between different layers in a neural network (Gu et al., 2018). They allow for the most common patterns learned on the bottom layers to be reused by the top layers in order to maintain a connection between different data representations along the whole network. Skip connections turned out to be crucial for deep neural network efficiency in recent studies (Iglovikov and Shvets, 2018). For RainNet, we use skip connections for the transition of learned patterns from the encoder to the decoder branch at the different resolutional levels.

One of the prerequisites for U-Net based architectures is that the spatial extent of input data has to be a multiple of 2^{n+1}, where n is the number of max pooling layers. As a consequence, the spatial extent on different resolutional levels becomes identical for the decoder and encoder branches. Correspondingly, the radar composite grids were transformed from the native spatial extent of 900×900 cells to the extent of 928×928 cells using mirror padding.

RainNet takes four consecutive radar composite grids as separate input channels (*t-15*, *t-10*, *t-5* minutes, and *t*, where *t* is the time of the nowcast) to produce a nowcast at time *t+5* minutes. Each grid contains 928×928 cells with an edge length of 1 km; for each cell, the input value is the logarithmic precipitation depth as retrieved from the radar-based precipitation product. There are five almost symmetrical resolutional levels for both decoder and encoder which utilize precipitation patterns at the full spatial input resolution of (*x, y*), at a quarter resolution (*x/2, y/2*), at (*x/4, y/4*), (*x/8, y/8*), and (*x/16, y/16*) respectively. To increase the robustness and to prevent overfitting of pattern representations at coarse resolutions, we implemented a dropout regularization technique (Srivastava et al., 2014). Finally, the output layer of resolution (*x, y*) with a linear activation function provides the predicted logarithmic precipitation (in mm) in each grid cell for *t+5* minutes.

RainNet differs fundamentally from ConvLSTM (Shi et al., 2015), a prior neural-network approach, which accounts for both spatial and temporal structures in radar data by using stacked convolutional as well as LSTM layers that preserve the spatial resolution of the input data alongside all the computational layers. LSTM networks have been observed to be brittle; in several application domains, convolutional neural networks have turned out to be numerically more stable during training, and make more accurate predictions than these recurrent neural networks (e.g., Bai et al., 2018; Gehring et al., 2017).

Therefore, RainNet uses a fully convolutional architecture, and does not use LSTM layers to propagate information through time. In order to make predictions with a larger lead time, we apply RainNet recursively. After predicting the estimated log-

precipitation for *t+5* minutes, the measured values for *t-10*, *t-5*, and *t* as well as the estimated value for *t+5* serve as the next input volume which yields the estimated log-precipitation for *t+10* minutes. The input window is then moved on incrementally.

2.2 Optimization procedure

In total, RainNet has almost 31.4 million parameters. We optimized these parameters using a procedure of which we show one iteration in Fig. 2: first, we read a sample of input data that consists of radar composite grids at time *t-15*, *t-10*, *t-5* minutes, and *t*, and a sample of the observed precipitation at time *t+5*. For both, input and observation, we increase the spatial extent to 928×928 using mirror padding, and transform precipitation depth x (in mm / 5 minutes) as follows (Eq. 1):

$$x_{transformed} = \ln(x_{raw} + 0.01) \tag{1}$$

Second, RainNet carries out a prediction based on the input data. Third, we calculate a loss function that represents the deviation between prediction and observation. Previously, Chen et al. (2018) showed that using the *logcosh* loss function is beneficial for the optimization of variational auto-encoders (VAE) in comparison to mean squared error. Accordingly, we employed the *logcosh* loss function as follows (Eq. 2):

$$Loss = \frac{\sum_{i=1}^{n} \ln(\cosh(now_i - obs_i))}{n} \tag{2}$$

$$\cosh(x) = \frac{1}{2}(e^x + e^{-x}) \tag{3}$$

where now_i and obs_i are nowcast and observation at the *i*-th location, respectively; $\cosh$ is the hyperbolic cosine function (Eq. 3); n is the number of cells in the radar composite grid.

Fourth, we update RainNet's model parameters to minimize the loss function using backpropagation algorithm where the Adam optimizer is utilized to compute the gradients (Kingma and Ba, 2015).

We optimized RainNet's parameters using 10 epochs (one epoch ends when the neural network saw every input data sample once, then the next epoch begins) with a mini batch of size 2 (one mini batch holds a few input data samples). The optimization procedure has converged on the 8^{th} epoch showing saturation of RainNet's performance on the validation data. The learning rate of the Adam optimizer had a value of 1e-04, while other parameters had default values from the original paper of Kingma and Ba (2015).

The entire setup was empirically identified as the most successful in terms of RainNet performance on validation data, while other configurations with different loss functions (e.g., mean absolute error, mean squared error) and optimization algorithms (e.g., stochastic gradient descent) have also converged. The average training time on a single GPU (NVIDIA GTX GeForce 1080Ti, NVIDIA GTX TITAN X, or NVIDIA Tesla P100) varies from 72 to 76 hours.

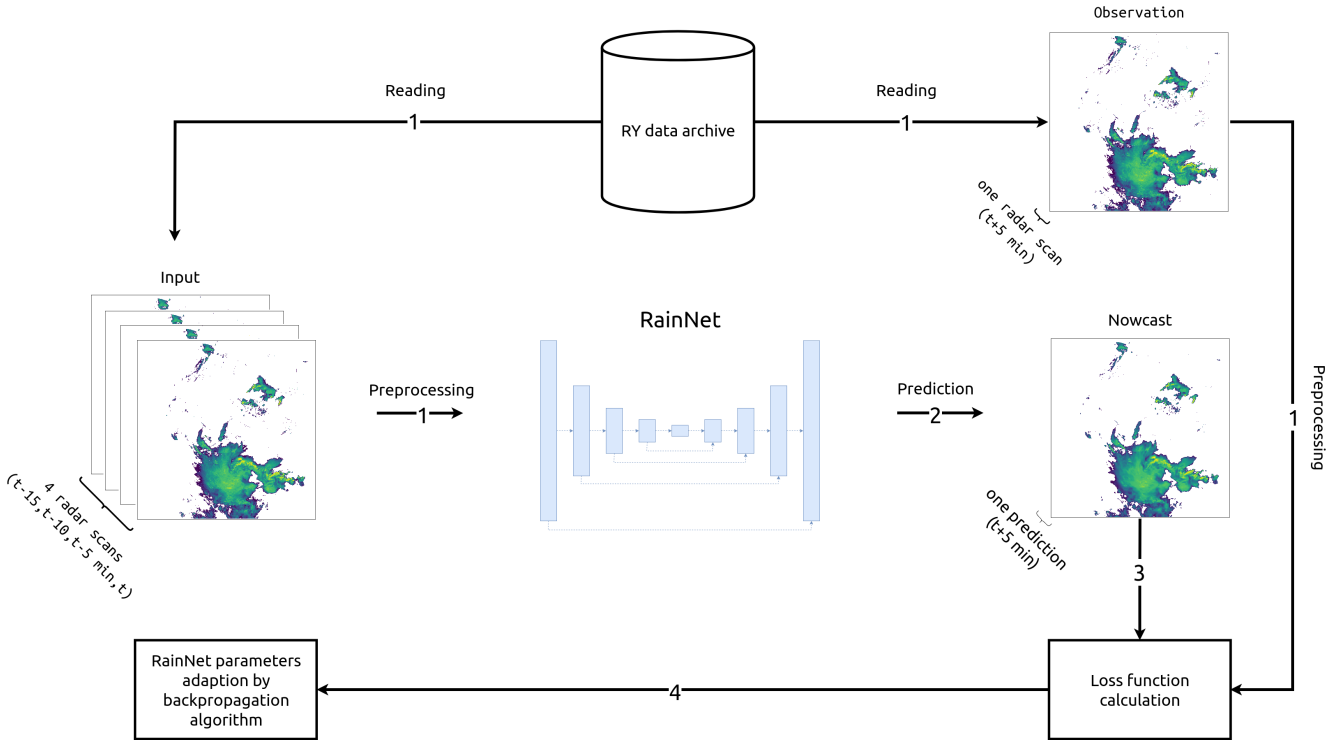

Figure 2. Illustration of one iteration step of the RainNet parameters optimization procedure.

We support this paper by a corresponding repository on GitHub (https://github.com/hydrogo/rainnet; Ayzel, 2020a) which holds the RainNet model architecture written in Python 3 programming language (https://python.org, last access: 28 January 2020) using the *Keras* deep learning library (Chollet et al., 2015) alongside its parameters (Ayzel, 2020b) which had been optimized on the radar data set which is described in the following section.

3 Data and experimental setup

3.1 Radar data

We use the RY product of the German Weather Service (DWD) as input data for training and validating the RainNet model. The RY product represents a quality-controlled rainfall-depth composite of 17 operational DWD Doppler radars. It has a spatial extent of 900×900 km, covers the whole area of Germany, and is available since 2006. The spatial and temporal resolution of the RY product is 1×1 km and 5 minutes, respectively.

In this study, we use RY data that covers the period from 2006 to 2017. We split the available RY data as follows: while we use data from 2006 to 2013 to optimize RainNet's model parameters and data from 2014 to 2015 to validate RainNet performance, data from 2016 to 2017 is used for model verification (Sect. 3.3). For both optimization and validation periods,

we keep only data from May to September and ignore time steps for which the precipitation field (with rainfall intensity more than 0.125 mm h^{-1}) covers less than 10% of the RY domain. For each subset of the data – for optimization, validation, and verification –, every time step (or frame) is used once as t_0 (forecast time) so that the resulting sequences that are used as input to a single forecast (t_0 - *15* min, ..., t_0) overlap in time. The number of resulting sequences amounts to 41988 for the optimization, 5722 for the validation, and 9626 for the verification (see also Sect 3.3).

3.2 Reference models

We use a nowcasting models from the *rainymotion* Python library (Ayzel et al., 2019) as benchmarks against which we evaluate RainNet. As the first baseline model, we use Eulerian persistence (further referred to as Persistence), which assumes that for any lead time *n* (minutes), precipitation at *t+n* is the same as at forecast time *t*. Despite its simplicity, it is quite a powerful model for very short lead times, which also establishes a solid verification efficiency baseline which can be achieved with a trivial model without any explicit assumptions. As the second baseline model, we use the Dense model from the *rainymotion* library (further referred to as Rainymotion) which is based on optical flow techniques for precipitation field tracking and the constant-vector advection scheme for precipitation field extrapolation. Ayzel et al. (2019) showed that this model has an equivalent or even superior performance in comparison to the operational RADVOR model from DWD for a wide range of rainfall events.

3.3 Verification experiments and performance evaluation

For benchmarking RainNet's predictive skill in comparison to the baseline models, Rainymotion and Persistence, we selected 11 events during the summer months of the verification period (2016–2017). These events are selected for covering a range of event characteristics with different rainfall intensity, spatial coverage, and duration. A detailed account of the events' properties was given by Ayzel et al. (2019).

We use three metrics for model verification: mean absolute error (MAE), critical success index (CSI), and fractions skill score (FSS). Each metric represents a different category of scores. MAE (Eq. 4) corresponds to the continuous category and maps the differences between nowcast and observed rainfall intensities; CSI (Eq. 5) is a categorical score which is based on a standard contingency table for calculating matches between Boolean variables which indicate the exceedance of specific rainfall intensity thresholds; FSS (Eq. 6) represents neighborhood verification scores and is based on comparing nowcast and observed fractional coverage of rainfall intensities exceeding specific thresholds in spatial neighborhoods (windows) of certain size.

$$MAE = \frac{\sum_{i=1}^{n} |now_i - obs_i|}{n} \tag{4}$$

$$CSI = \frac{hits}{hits + false\ alarms + misses} \tag{5}$$

$$FSS = 1 - \frac{\sum_{i=1}^{n}(P_n - P_o)^2}{\sum_{i=1}^{n} P_n^2 + \sum_{i=1}^{n} P_o^2} \quad (6)$$

where quantities now_i and obs_i are nowcast and observed rainfall rate in the *i*-th pixel of the corresponding radar image and n is the number of pixels. *Hits*, *false alarms*, and *misses* are defined by the contingency table and the corresponding threshold value. Quantities P_n and P_o represent the nowcast and observed fractions of rainfall intensities exceeding a specific threshold for a defined neighborhood size, respectively. MAE is positive and unbounded with a perfect score of 0; both CSI and FSS can vary from 0 to 1 with a perfect score of 1. We have applied threshold rain rates of 0.125, 1, ~~and~~ 5, 10, and 15 mm h^{-1} for calculating ~~CSI and~~ the CSI and the FSS. For calculating the FSS we use neighborhood (window) sizes of 1, 5, 10, and 20 km.

The verification metrics we use in this study quantify the models' performance from different perspectives. The MAE captures errors in rainfall rate prediction (the less the better), CSI (the higher the better) captures model accuracy – the fraction of the forecast event that was correctly predicted – but it does not distinguish between the sources of errors. The FSS determines how the nowcast skill depends on both threshold of rainfall exceedance and spatial scale (Mittermaier and Roberts, 2010).

In addition to standard verification metrics described above, we calculate the power spectral density (PSD) of nowcasts and corresponding observations using Welch's method (Welch, 1967) to investigate the effects of smoothing demonstrated by different models.

4 Results and discussion

For each event, RainNet was used to compute nowcasts at lead times from 5 to 60 minutes (in 5-minute steps). To predict the precipitation at time *t+5* minutes (*t* being forecast time), we used the four latest radar images (at time *t-15*, *t-10*, *t-5* minutes, and *t*) as input. And since RainNet was only trained to predict precipitation at five minutes lead time, predictions beyond *t+5* were made recursively: in order to predict precipitation at *t+10*, we considered the prediction at *t+5* as the latest observation. That recursive procedure was repeated up to a maximum lead time of 60 minutes. Rainymotion uses the two latest radar composite grids (*t-5*, *t*) in order to retrieve a velocity field, and then to advect the latest radar-based precipitation observation at forecast time *t* to *t+5*, *t+10*, ..., and *t+60*.

Fig. 3 shows the routine verification metrics MAE and CSI for RainNet, Rainymotion, and Persistence, as a function of lead time. Preliminary analysis had shown the same general pattern of model efficiency for each of the eleven events (Sect. S1 in the Supplement), which is why we only show the average metrics over all events. ~~Clearly, RainNet~~ The results basically fall into two groups:

The first group includes the MAE and the CSI metrics up to a threshold of 5 mm h^{-1}. For these, RainNet clearly outperforms the benchmarks ~~in each metric and~~ at any lead time (differences between models were tested to be significant with the two-tailored T-test at a significance level of 5%, results not shown). Persistence is the least skillful, as could be expected for a trivial baseline. The relative differences between RainNet and Rainymotion are more pronounced for the MAE than for the CSI. For the MAE, the ~~difference between RainNet and~~ advance of RainNet over Rainymotion increases with lead time. For the CSI,

the ~~difference between RainNet and~~ superiority of RainNet over Rainymotion appears to be highest for intermediate lead times between 20 and 40 minutes. The performance of all models, in terms of CSI, decreases with increasing intensity thresholds. ~~The relative difference between RainNet and Rainymotion metrics is the lowest for the CSI at~~

That trend – a decreasing CSI with increasing intensity – continues with the second group of metrics: the CSI for thresholds of 10 and 15 mm h^{-1}. For both metrics and any of the competing methods at any lead time, the CSI does not exceed a value of 0.31 (obtained by RainNet at five minutes lead time and a threshold of 10 mm h^{-1}). That is below a value of $1/e \approx 0.37$ which had been suggested by Germann and Zawadzki (2002) as a "limit of predictability" (under the assumption that the optimal value of the metric is 1 and that it follows an exponential-like decay over lead time). Irrespective of such an – admittedly arbitrary – predictability threshold, the loss of skill from an intensity threshold of 5 to 10 mm h^{-1} is remarkable – for all competing models. Visually more apparent, however, is another property of the second group of metrics, which is that Rainymotion outperforms RainNet (except for a threshold of 10 mm h^{-1} at a lead time of 5 and 60 minutes). That becomes most pronounced for the CSI at 15 mm h^{-1}): while RainNet has a similar CSI value as Rainymotion at a lead time of five minutes, it entirely fails at predicting the exceedance of 15 mm h^{-1}) for longer lead times.

In summary, Fig. 3 suggests that RainNet outperforms Rainymotion (as a representative of standard tracking and extrapolation techniques based on optical flow) for ~~any of the metrics shown. Yet, the CSI for a threshold of~~ low and intermediate rain rates (up to 5 mm h^{-1}). Neither RainNet nor Rainymotion appear to have much skill at predicting the exceedance of 10 mm h^{-1} ~~already implies that RainNet might have~~ , but the loss of skill for high intensities is particularly remarkable for RainNet which obviously has difficulties in predicting pronounced precipitation features with high intensities.

In order to better understand the fundamental properties of RainNet predictions in contrast to Rainymotion, we continue by inspecting a nowcast at three different lead times (5, 30 and 60 minutes), for a verification event at an arbitrarily selected forecast time (2016-05-29 19:15:00 UTC). The top row of Fig. 4 shows the observed precipitation, the second and third row show Rainymotion and RainNet predictions. And since it is visually challenging to track the motion pattern at the scale of 900×900 km by eyeball, we illustrate the velocity field as obtained from optical flow which forms the basis for Rainymotion's prediction. While it is certainly difficult to infer the predictive performance of the two models from this figure, another feature becomes immediately striking: RainNet introduces a spatial smoothing which appears to substantially increase with lead time. In order to quantify that visual impression, we calculated, for the same example, the power spectral density (PSD) of the nowcasts and the corresponding observations (bottom row in Fig. 4), using Welch's method (Welch, 1967). In simple terms, the PSD represents the prominence of precipitation features at different spatial scales, expressed as the spectral power at different wavelengths after a two-dimensional Fast Fourier Transform. The power spectrum itself is not of specific interest here; it is the loss of power at different length scales, relative to the observation, that is relevant in this context. The loss of power of Rainymotion nowcasts appears to be constrained to spatial scales below 4 km, and does not seem to depend on lead time (see also Ayzel et al., 2019). For RainNet, however, a substantial loss of power at length scales below 16 km becomes apparent at a lead time of 5 minutes. For longer lead times of 30 and 60 minutes, that loss of power grows and propagates to scales of up to 32 km. That loss of power over a range of scales corresponds to our visual impression of spatial smoothing.

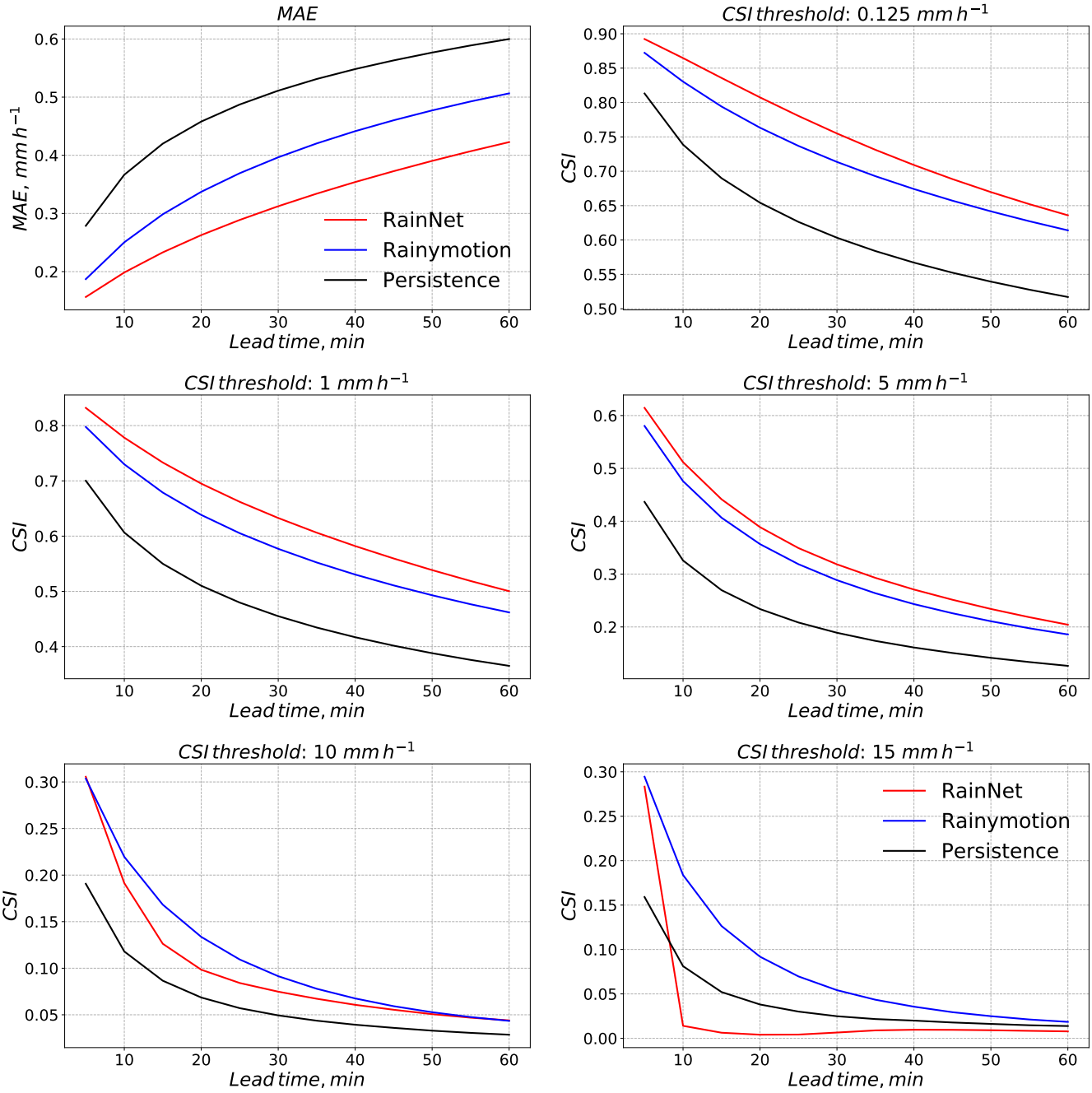

Figure 3. Mean Absolute Error (MAE) and Critical Success Index (CSI) for three different intensity thresholds (0.125 mm h^{-1}, 1 mm h^{-1}, 5 mm h^{-1}, 10 mm h^{-1}, 15 mm h^{-1}). The metrics are shown as a function of lead time. All values represent the average of the corresponding metric over all 11 verification events.

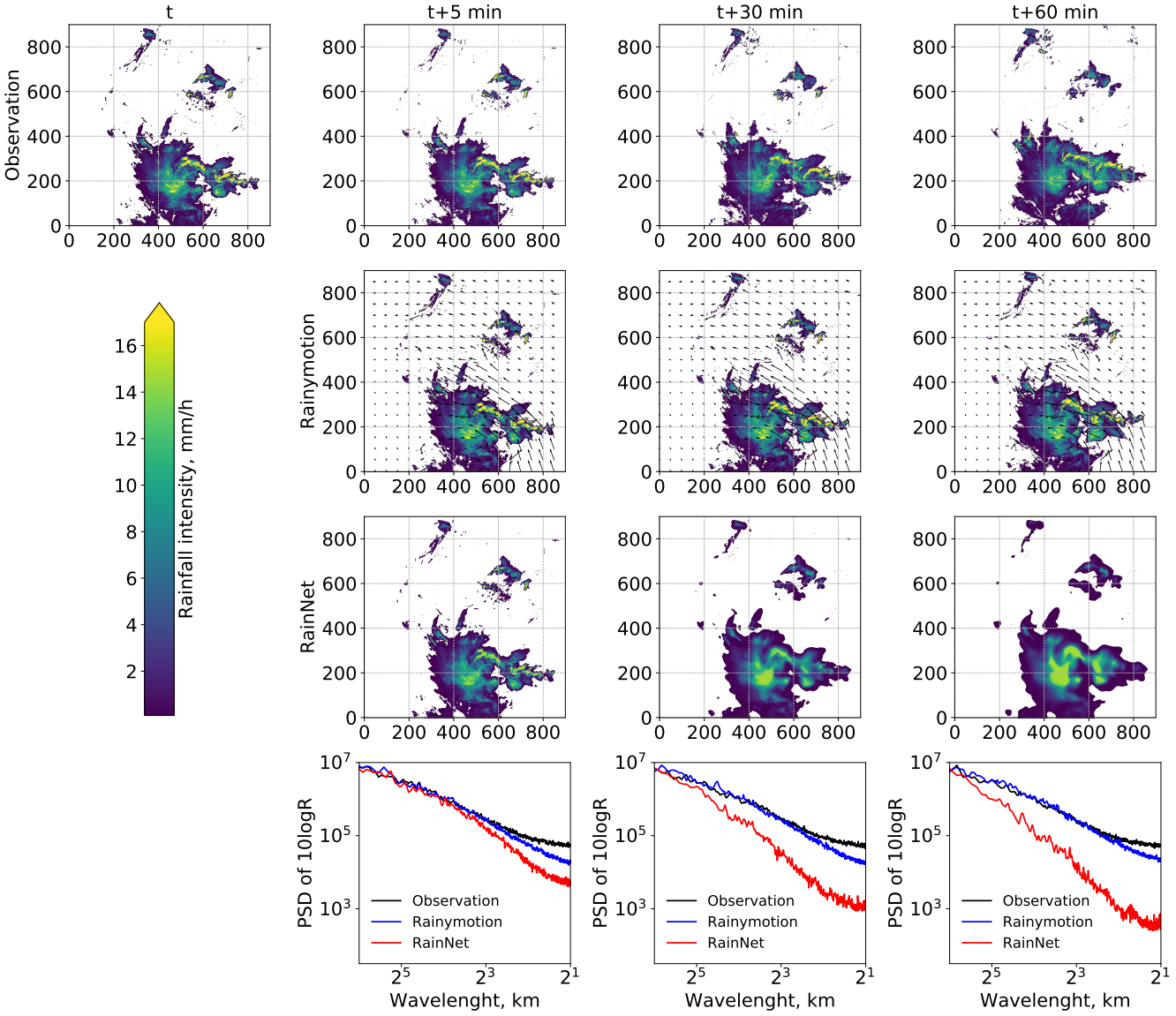

Figure 4. Precipitation observations as well as Rainymotion and RainNet nowcasts at *t*=2016-05-29 19:15; *top row*: observed precipitation intensity at time *t*, *t+5*, *t+30* and *t+60* minutes; *second row*: corresponding Rainymotion prediction, together with the underlying velocity field obtained from optical flow; *bottom row*: power spectral density plots for observations and nowcasts at lead times 5, 30 and 60 minutes.

In order to investigate whether that loss of spectral power at smaller scales is a general property of RainNet predictions, we computed the PSD for each forecast time in each verification event, in order to obtain an average PSD for observations and nowcasts, at lead times of 5, 30, and 60 minutes. The corresponding results are shown in Fig. 5. They confirm that the behaviour observed in the bottom row of Fig. 4 is, in fact, representative for the entirety of verification events. Precipitation

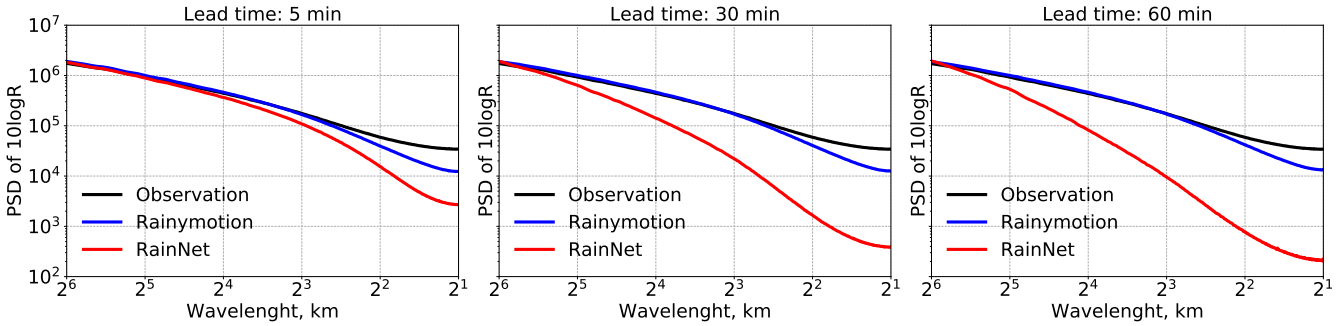

Figure 5. PSD averaged over all verification events and nowcasts, for lead times of 5, 30, and 60 minutes.

fields predicted by RainNet are much smoother than both the observed fields and the Rainymotion nowcasts. At a lead time of five minutes, RainNet starts to lose power at a scale of 16 km. That loss accumulates over lead time and becomes effective up to a scale of 32 km at a lead time of 60 minutes. These results confirm qualitative findings of Shi et al. (2015) and Shi et al. (2018) who described their nowcasts as "smooth" or "fuzzy".

RainNet obviously learned, as the optimal way to minimize the loss function, to introduce a certain level of smoothing for the prediction at time *t+5* minutes. It might even have learned to systematically "attenuate" high intensity features as a strategy to minimize the loss function – which would be consistent with the results for the CSI at a threshold of 15 mm h^{-1}, as shown in Fig. 3. For the sake of simplicity, though, we will refer to the overall effect as "smoothing" in the rest of the paper. According to the loss of spectral power, the smoothing is still small at a length scale of 16 km, but becomes increasingly effective at smaller scales from 8 to 2 km. It is important to note that the loss of power below length scales of 16 km at a lead time of five minutes is an essential property of RainNet. It reflects the learning outcome, and illustrates how RainNet factors in predictive uncertainty at five minutes lead time by smoothing over small spatial scales. Conversely, the increasing loss of power and its propagation to larger scales up to 32 km are *not* an inherent property of RainNet, but a consequence of its recursive application in our study context: as the predictions at short lead times serve as model input for predictions at longer lead times, the results become increasingly smooth. So while the smoothing introduced at five minutes lead time can be interpreted as a direct result of the learning procedure, the cumulative smoothing at longer lead time has to be considered rather an artifact, similar to the effect of "numerical diffusion" in numerically solving the advection equation.

Given this understanding of RainNet's properties, we used the Fractions Skill Score (FSS) to provide further insight into the dependency of predictive skill on the spatial scale. To that end, the FSS was obtained by comparing the predicted and observed fractional coverage of pixels (inside a spatial window / neighbourhood) that exceed a certain intensity threshold (see Eq. 6 in Sect. 3.3). Fig. 6 shows the FSS for Rainymotion and RainNet, as an average over all verification events, for spatial window sizes of 1, 5, 10, and 20 km, and for intensity thresholds of 0.125, 1, ~~and~~ 5, 10 and 15 mm h^{-1}. In addition to the color code, the value of the FSS is given for each combination of window size (scale) and intensity. In case one model is superior to the other, the correspondingly higher FSS value is highlighted in bold black digits.

Based on the above results and discussion of RainNet's versus Rainymotion's predictive properties, the FSS figures are plausible, and provide a more formalized approach to express ~~the effects of smoothing~~ different behaviour of RainNet and Rainymotion in terms of predictive skill. ~~For a window size of 1 km – i.e., the native grid resolution – RainNet outperforms Rainymotion for each intensity threshold and lead time. That finding is consistent with the CSI as shown in Fig. 3. Yet, the superiority of RainNet successively becomes lost – and even reversed – with increasing~~ In general, the skill of both models decreases with decreasing window sizes, ~~intensity thresholds, and lead times. That effect becomes most pronounced at~~ increasing lead times, and increasing intensity thresholds. RainNet tends to outperform Rainymotion at lower rainfall intensities (up to 5 mm h^{-1}) at the native grid resolution (i.e. a window size of ~~20 km, an intensity~~ 1 km). With increasing window sizes and intensity thresholds, Rainymotion becomes the superior model. At an intensity threshold of 5 mm h^{-1}, ~~and a lead time of 60 minutes, where~~ Rainymotion outperforms RainNet ~~by an FSS of 0.64 versus 0.54, the largest difference found in the entire set of FSS values in Fig. 6. Yet, Rainymotion already starts to slightly outperform RainNet at a window size of~~ at window sizes equal or greater than 5 km~~and an intensity threshold of 5~~ . At intensity thresholds of 10 and 15 mm h^{-1}~~(all lead times), or~~ , Rainymotion is superior at any lead time and window size (except a window size of ~~10 km and a lead time of 60 minutes (all intensity thresholds~~1 km for a threshold of 10 mm h^{-1}).

The dependency of the FSS (or, rather, the difference of FSS values between Rainymotion and RainNet) on spatial scale, intensity threshold, and lead time, is a direct result of inherent model properties. Rainymotion advects precipitation features, but preserves their intensity~~, while RainNet has not only learned how precipitation fields move in space, but that smoothing in space is an efficient way to minimize the loss function~~. When we increase the size of the spatial neighbourhood around a pixel, this neighbourhood could, at some size, include high intensity precipitation features that Rainymotion has preserved, but ~~just~~ slightly misplaced. RainNet's loss function, however, ~~could have lost such features entirely as a result of smoothing~~only accounts for the native grid at 1 km resolution, so it has no notion of what could be a slight or "acceptable" displacement error. Instead, RainNet has learned spatial smoothing as an efficient way to factor in spatial uncertainty and minimize the loss function, resulting into a loss of high intensity features. As discussed above, that effect becomes increasingly prominent for longer lead times because the effect of smoothing propagates.

5 Summary and conclusions

In this study, we have presented RainNet, a deep convolutional neural network architecture for radar-based precipitation nowcasting. Its design was inspired by the U-Net and SegNet families of deep learning models for binary segmentation, and follows an encoder-decoder architecture in which the encoder progressively downscales the spatial resolution using pooling, followed by convolutional layers; and the decoder progressively upscales the learned patterns to a higher spatial resolution using upsampling, followed by convolutional layers.

RainNet was trained to predict precipitation at a lead time of five minutes, using several years of quality-controlled weather radar composites based on the DWD weather radar network. That data covers Germany with a spatial domain of 900×900 km, and has a resolution of 1 km in space and 5 minutes in time. Independent verification experiments were carried out on

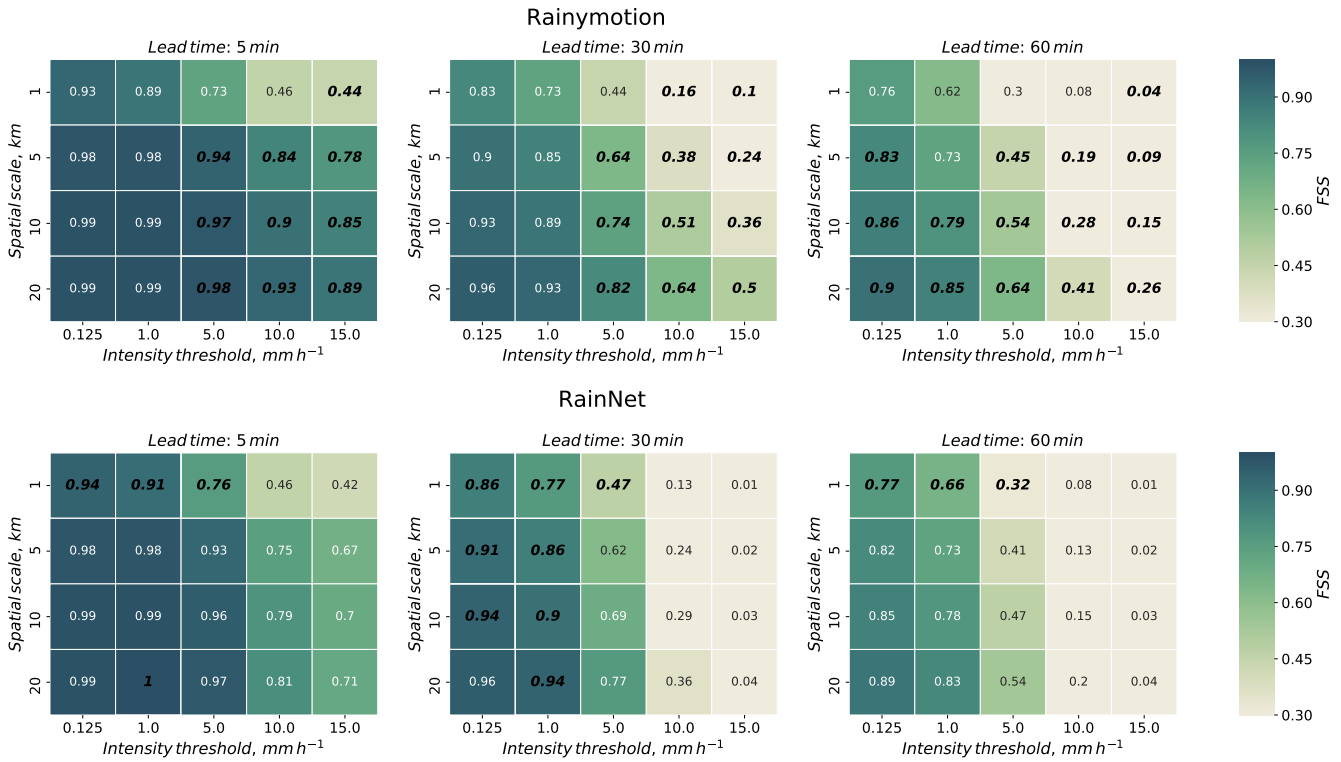

Figure 6. Fractions Skill Score (FSS) for Rainymotion (*top panel*) and RainNet (*bottom panel*), for 5, 30, and 60 minutes lead time, and spatial window sizes of 1, 5, 10 and 20 km, and for intensity thresholds of 0.125, 1, ~~and~~ 5, 10 and 15 mm h^{-1}. In addition to the color code of the FSS, we added the numerical FSS values. The FSS value of the model which is significantly superior for a specific combination of window size, intensity threshold, and lead time is typed in bold black digits, for the inferior model in regular.

eleven summer precipitation events from 2016 to 2017. In order to achieve a lead time of 60 minutes, a recursive approach was implemented by using RainNet predictions at five minutes lead time as model input for longer lead times. In the verification experiments, ~~Eurlerian~~ Eulerian persistence served as a trivial benchmark. As an additional benchmark, we used a model from the *rainymotion* library which had previously been shown to outperform the operational nowcasting model of the German Weather Service for the same set of verification events.

RainNet significantly outperformed both benchmark models at all lead times up to 60 minutes for the routine verification metrics Mean Absolute Error (MAE) and the Critical Success Index (CSI~~,~~) at intensity thresholds of 0.125, 1, and 5 mm h^{-1}~~)~~. Depending on the verification metric, ~~the results~~ these results would correspond to an extension of the effective lead time in the order of 10–20 minutes by RainNet as compared to Rainymotion. ~~Since both Rainymotion and RainNet substantially outperformed the trivial benchmark of Persistence, the latter was not considered in subsequent analyses~~However, Rainymotion turned out to be clearly superior in predicting the exceedance of higher intensity thresholds (here 10 and 15 mm h^{-1}), as shown by the corresponding CSI analysis.

~~Apart from its superiority in terms of MAE and CSI, a remarkable property of RainNetpredictions, in comparison to~~ ~~Rainymotion, is the~~ RainNet's limited ability to predict high rainfall intensities could be attributed to a remarkable level of spatial smoothing in its predictions. That smoothing becomes increasingly apparent at longer lead times. Yet, it is already prominent at a lead time of five minutes. That was confirmed by an analysis of Power Spectral Density which showed, at time *t+5* minutes, a loss of spectral power at length scales of 16 km and below. Obviously, RainNet has learned an optimal level of smoothing to produce a nowcast at 5 minutes lead time. In that sense, the loss of spectral power at small scales is informative, as it reflects the limits of predictability as a function of spatial scale. Beyond the lead time of five minutes, however, the increasing level of smoothing is a mere artifact – an analogue to numerical diffusion – that is not a property of RainNet itself, but of its recursive application: as we repeatedly use smoothed nowcasts as model inputs, we cumulate the effect of smoothing over time. That certainly is an undesirable property, and it becomes particularly unfavourable for the prediction of high-intensity precipitation features. As was shown on the basis of the Fractions Skill Score (FSS), Rainymotion outperforms RainNet ~~for intensive precipitation (>~~already at an intensity of 5 mm h^{-1}) once we start to evaluate the performance in a spatial neighbourhood around the native grid pixel of 1×1 km size. This is because Rainymotion preserves distinct precipitation features, but tends to misplace them. RainNet, however, tends to lose such features over longer lead times due to cumulative smoothing effects – more so if it is applied recursively.

From an early warning perspective, that property of RainNet clearly limits its usefulness. There are, however, options to address that issue in future research:

- The loss function used in the training could be adjusted in order to penalize the loss of power at small spatial scales. The loss function explicitly represents our requirements to the model. Verifying the model by other performance metrics will typically reveal whether these metrics are rather in agreement or in conflict with these requirements. In our case, the *logcosh* loss function appears to favour a low MAE, but at the cost of losing distinct precipitation features. In general, future users need to be aware that, apart from the network design, the optimization itself constitutes the main difference to "heuristic" tracking-and-extrapolation techniques (such as Rainymotion) which do not use any systematic parameter optimization. The training procedure will stubbornly attempt to minimize the loss function, irrespective of what researchers consider as "physically plausible". For many researchers in the field of nowcasting, that notion might be in stark contrast to experiences with "conventional" nowcasting techniques which tend to effortlessly produce at least plausible patterns;

- RainNet should be directly trained to predict precipitation at lead times beyond five minutes. However, preliminary training experiments with that learning task had difficulties to converge. We thus recommend to still use recursive predictions as *model input* for longer lead times during training, in order to improve convergence. For example, to predict precipitation at time *t+10* minutes, RainNet could be trained using precipitation at time *t-15*, *t-10*, ..., *t* minutes as input, but using the recursive prediction at time *t+5* as an *additional* input layer, too. But while the direct prediction of precipitation at longer lead times should reduce excessive smoothing as a result of numerical diffusion, we would still expect the level of smoothing to increase with lead time, as a result of the predictive uncertainty at small scales;

– As an alternative to predict continuous values of precipitation intensity, RainNet could be trained to predict the exceedance of specific intensity thresholds instead. That would correspond to a binary segmentation task. It is possible that the objective of learning the segmentation for *low* intensities might be in conflict with learning it for *high* intensities. That is why the training could be carried out both separately and jointly for disparate thresholds, in order to investigate whether there are inherent trade-offs. From an early warning perspective, it makes sense to train RainNet for binary segmentation based on user defined thresholds that are governed by the context of risk management. The additional advantage of training RainNet to predict threshold exceedance is that we could use its output directly as a measure of uncertainty (of that exceedance).

We consider any of those options worth being pursued in order to increase the usefulness of RainNet in an early warning context – i.e. to better represent precipitation intensities that exceed hazardous thresholds –, and we would expect the overall architecture of RainNet to be a helpful starting point.

Yet, the key issue of precipitation prediction – the anticipation of convective initialization as well as the growth and dissipation of precipitation in the imminent future – still appears to be unresolved. It is an inherent limitation of nowcasting models purely based on optical flow: they can extrapolate motion fairly well, but they cannot predict intensity dynamics. Deep learning architectures, however, *might* be able to learn recurrent patterns of growth and dissipation, although it will be challenging to verify if they actually *did*. In the context of this study, though, we have to assume that RainNet has rather learned the representation of motion patterns instead of rainfall intensity dynamics: for a lead time of five minutes, the effects of motion can generally be expected to dominate over the effects of intensity dynamics, which will propagate to the learning results. The fact that we actually could recursively use the RainNet predictions at five minutes lead time in order to predict precipitation at one hour lead time also implies that RainNet, in essence, learned to represent motion patterns and optimal smoothing. In that case, the trained model might even be applicable on data in another region which could be tested in future verification experiments.

Another limitation in successfully learning patterns of intensity growth and dissipation might be the input data itself. While we do not exclude the possibility that such patterns could be learned from just two-dimensional radar composites, other input variables might add essential information on imminent atmospheric dynamics – the predisposition of the atmosphere to produce or to dissolve precipitation. Such additional data might include 3-dimensional radar volume data, dual-pol radar moments, or output fields of numerical weather prediction (NWP) models. Formally, the inclusion of NWP fields in a learning framework could be considered as a different way of assimilation, combining – in a data-driven way – the information content of physical models and observations.

Our study provides, after Shi et al. (2015, 2017, 2018), another proof-of-concept that convolutional neural networks provide a firm basis to compete with conventional nowcasting models based on optical flow (most recently, Google Research has also reported similar attempts based on a U-Net architecture, see Agrawal et al. (2019)). Yet, this study should rather be considered as a starting point: to further improve the predictive skill of convolutional neural networks, and to better understand the properties of their predictions – in a statistical sense, but also in how processes of motion and intensity dynamics are reflected. To that end, computational complexity and the cost of the training process still have to be considered as inhibitive, despite the tremendous progress achieved in the past years: RainNet's training would require almost a year on a standard

desktop CPU, in contrast to the three days on a modern desktop GPU (although the latter is a challenge to implement for non-experts). Yet, it is possible to run deep learning models with already optimized (pretrained) weights on a desktop computer. Thus, it is important not only to make available the code of the network architecture, but also the corresponding weights, applicable using open source software tools and libraries. We provide all this – code, pretrained weights, as well as training and verification data – as an input to future studies on open repositories (Ayzel, 2020a, b, c).

Code and data availability. The RainNet model is free and open source. It is distributed under the MIT software license which allows unrestricted use. The source code is provided through a GitHub repository https://github.com/hydrogo/rainnet (last access: 30 January 2020); a snapshot of Rainnet v1.0 is also available at http://doi.org/10.5281/zenodo.3631038 (Ayzel, 2020a); the pretrained RainNet model and its weights are available at http://doi.org/10.5281/zenodo.3630429 (Ayzel, 2020b). DWD provided the sample data of the RY product; it is available at http://doi.org/10.5281/zenodo.3629951 (Ayzel, 2020c).

Author contributions. GA developed the RainNet model, carried out the benchmark experiments, and wrote the manuscript; TS and MH supervised the study and co-authored the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. GA was financially supported by Geo.X, the Research Network for Geosciences in Berlin and Potsdam (Project-number: SO_087_GeoX). GA would like to thank Open Data Science community (ods.ai) for many valuable discussions and educational help in the growing field of deep learning. We ran our experiments using GPU computation resources of the Machine Learning Group of the University of Potsdam (Potsdam, Germany) and the Shared Facility Center "Data Center of FEB RAS" (Khabarovsk, Russia). We acknowledge the support of Deutsche Forschungsgemeinschaft (German Research Foundation) and the Open Access Publication Fund of Potsdam University.

References

Agrawal, S., Barrington, L., Bromberg, C., Burge, J., Gazen, C., and Hickey, J.: Machine Learning for Precipitation Nowcasting from Radar Images, https://arxiv.org/abs/1912.12132, last access: 28 January 2020, 2019.

Austin, G. L. and Bellon, A.: The use of digital weather radar records for short-term precipitation forecasting, Quarterly Journal of the Royal Meteorological Society, 100, 658–664, https://doi.org/10.1002/qj.49710042612, http://doi.wiley.com/10.1002/qj.49710042612, 1974.

Ayzel, G.: hydrogo/rainnet: RainNet v1.0-gmdd, https://doi.org/10.5281/zenodo.3631038, 2020a.

Ayzel, G.: RainNet: pretrained model and weights, https://doi.org/10.5281/zenodo.3630429, 2020b.

Ayzel, G.: RYDL: the sample data of the RY product for deep learning applications, https://doi.org/10.5281/zenodo.3629951, 2020c.

Ayzel, G., Heistermann, M., and Winterrath, T.: Optical flow models as an open benchmark for radar-based precipitation nowcasting (rainymotion v0.1), Geoscientific Model Development, 12, 1387–1402, https://doi.org/10.5194/gmd-12-1387-2019, https://www.geosci-model-dev.net/12/1387/2019/, 2019.

Badrinarayanan, V., Kendall, A., and Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 2481–2495, https://doi.org/10.1109/TPAMI.2016.2644615, 2017.

Bai, S., Kolter, J. Z., and Koltun, V.: An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, https://arxiv.org/abs/1803.01271, last access: 28 January 2020, 2018.

Bauer, P., Thorpe, A., and Brunet, G.: The quiet revolution of numerical weather prediction, Nature, 525, 47–55, https://doi.org/10.1038/nature14956, http://www.nature.com/doifinder/10.1038/nature14956, 2015.

Boureau, Y.-L., Ponce, J., and LeCun, Y.: A Theoretical Analysis of Feature Pooling in Visual Recognition, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, pp. 111–118, Omnipress, Madison, WI, USA, 2010.

Chen, P., Chen, G., and Zhang, S.: Log Hyperbolic Cosine Loss Improves Variational Auto-Encoder, https://openreview.net/forum?id=rkglvsC9Ym, last access: 28 January 2020, 2018.

Chollet, F. et al.: Keras, https://keras.io, 2015.

Dahl, G. E., Sainath, T. N., and Hinton, G. E.: Improving deep neural networks for LVCSR using rectified linear units and dropout, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8609–8613, https://doi.org/10.1109/ICASSP.2013.6639346, 2013.

Dueben, P. D. and Bauer, P.: Challenges and design choices for global weather and climate models based on machine learning, Geoscientific Model Development, 11, 3999–4009, https://doi.org/10.5194/gmd-11-3999-2018, https://www.geosci-model-dev.net/11/3999/2018/, 2018.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N.: Convolutional Sequence to Sequence Learning, in: Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, pp. 1243–1252, JMLR.org, 2017.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G.: Could Machine Learning Break the Convection Parameterization Deadlock?, Geophysical Research Letters, 45, 5742–5751, https://doi.org/10.1029/2018GL078202, http://doi.wiley.com/10.1029/2018GL078202, 2018.

Germann, U. and Zawadzki, I.: Scale-Dependence of the Predictability of Precipitation from Continental Radar Images. Part I: Description of the Methodology, Monthly Weather Review, 130, 2859–2873, https://doi.org/10.1175/1520-0493(2002)130<2859:SDOTPO>2.0.CO;2, https://doi.org/10.1175/1520-0493(2002)130<2859:SDOTPO>2.0.CO;2, 2002.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., and Chen, T.: Recent advances in convolutional neural networks, Pattern Recognition, 77, 354–377, https://doi.org/10.1016/j.patcog.2017.10.013, http://www.sciencedirect.com/science/article/pii/S0031320317304120, 2018.

Iglovikov, V. and Shvets, A.: TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation, https://arxiv.org/abs/1801.05746, last access: 28 January 2020, 2018.

Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, edited by Bengio, Y. and LeCun, Y., http://arxiv.org/abs/1412.6980, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, in: Advances in Neural Information Processing Systems 25, edited by Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., pp. 1097–1105, Curran Associates, Inc., http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf, 2012.

LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning, Nature, 521, 436–444, https://doi.org/10.1038/nature14539, http://www.nature.com/articles/nature14539, 2015.

Lin, C., Vasić, S., Kilambi, A., Turner, B., and Zawadzki, I.: Precipitation forecast skill of numerical weather prediction models and radar nowcasts, Geophysical Research Letters, 32, https://doi.org/10.1029/2005GL023451, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005GL023451, 2005.

Long, J., Shelhamer, E., and Darrell, T.: Fully Convolutional Networks for Semantic Segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

Mittermaier, M. and Roberts, N.: Intercomparison of Spatial Forecast Verification Methods: Identifying Skillful Spatial Scales Using the Fractions Skill Score, Weather and Forecasting, 25, 343–354, https://doi.org/10.1175/2009WAF2222260.1, https://doi.org/10.1175/2009WAF2222260.1, 2010.

Nair, V. and Hinton, G. E.: Interpersonal Informatics: Making Social Influence Visible, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, pp. 807–814, Omnipress, Madison, WI, USA, 2010.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat: Deep learning and process understanding for data-driven Earth system science, Nature, 566, 195–204, https://doi.org/10.1038/s41586-019-0912-1, https://doi.org/10.1038/s41586-019-0912-1, 2019.

Reyniers, M.: Quantitative precipitation forecasts based on radar observations: Principles, algorithms and operational systems, Institut Royal Météorologique de Belgique, https://www.meteo.be/meteo/download/fr/3040165/pdf/rmi_scpub-1261.pdf, 2008.

Ronneberger, O., Fischer, P., and Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, edited by Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., pp. 234–241, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-319-24574-4_28, 2015.

Shi, E., Li, Q., Gu, D., and Zhao, Z.: A Method of Weather Radar Echo Extrapolation Based on Convolutional Neural Networks, in: MultiMedia Modeling, edited by Schoeffmann, K., Chalidabhongse, T. H., Ngo, C. W., Aramvith, S., O'Connor, N. E., Ho, Y.-S., Gabbouj, M., and Elgammal, A., pp. 16–28, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-319-73603-7_2, https://link.springer.com/chapter/10.1007%2F978-3-319-73603-7_2, 2018.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c.: Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, in: Advances in Neural Information Processing Systems 28, edited by Cortes, C.,

Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., pp. 802–810, Curran Associates, Inc., http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting.pdf, 2015.

Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c.: Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model, in: Advances in Neural Information Processing Systems 30, edited by Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., pp. 5617–5627, Curran Associates, Inc., http://papers.nips.cc/paper/7145-deep-learning-for-precipitation-nowcasting-a-benchmark-and-a-new-model.pdf, 2017.

Singh, S., Sarkar, S., and Mitra, P.: Leveraging Convolutions in Recurrent Neural Networks for Doppler Weather Radar Echo Prediction, in: Advances in Neural Networks - ISNN 2017, edited by Cong, F., Leung, A., and Wei, Q., pp. 310–317, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-319-59081-3_37, https://link.springer.com/chapter/10.1007%2F978-3-319-59081-3_37, 2017.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, J. Mach. Learn. Res., 15, 1929–1958, 2014.

Srivastava, R. K., Greff, K., and Schmidhuber, J.: Training Very Deep Networks, in: Advances in Neural Information Processing Systems 28, edited by Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., pp. 2377–2385, Curran Associates, Inc., http://papers.nips.cc/paper/5850-training-very-deep-networks.pdf, 2015.

Sun, J., Xue, M., Wilson, J. W., Zawadzki, I., Ballard, S. P., Onvlee-Hooimeyer, J., Joe, P., Barker, D. M., Li, P.-W., Golding, B., Xu, M., and Pinto, J.: Use of NWP for Nowcasting Convective Precipitation: Recent Progress and Challenges, Bulletin of the American Meteorological Society, 95, 409–426, https://doi.org/10.1175/BAMS-D-11-00263.1, https://doi.org/10.1175/BAMS-D-11-00263.1, 2014.

Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, in: Advances in Neural Information Processing Systems 27, edited by Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., pp. 3104–3112, Curran Associates, Inc., http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf, 2014.

Welch, P.: The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms, IEEE Transactions on audio and electroacoustics, 15, 70–73, 1967.

Wilson, J. W., Crook, N. A., Mueller, C. K., Sun, J., and Dixon, M.: Nowcasting Thunderstorms: A Status Report, Bulletin of the American Meteorological Society, 79, 2079–2099, https://doi.org/10.1175/1520-0477(1998)079<2079:NTASR>2.0.CO;2, https://journals.ametsoc.org/doi/abs/10.1175/1520-0477%281998%29079%3C2079%3ANTASR%3E2.0.CO%3B2, 1998.