Motion Field Forecasting and Ensemble Generation in Rainfall Nowcasting

A Data-Driven Approach

Daniel A. Blázquez Martín

To obtain the degree of Master of Science at the Delft University of Technology, to be defended publicly on 11-07-2024
An electronic version of this thesis is available at http://repository.tudelft.nl/

Student number:	5689775	
Project duration:	February 12, 2024 – July 11, 2024	
Thesis committee:	Dr. Ruben Imhoff	Deltares, supervisor
	Dr. M.A. (Marc) Schleiss	TU Delft, supervisor
	Dr. Riccardo Taormina	TU Delft, secondary supervisor

Deltares TU Delft

Abstract

This thesis aims to enhance rainfall nowcasting by improving motion field predictions and ensemble generation within PySTEPS using machine learning techniques. Accurate nowcasting is crucial for flood early warning, agriculture, transportation, and public safety. The steady-state assumptions and ensemble generation in PySTEPS, and many other nowcasting methods, face significant challenges in maintaining accuracy over time, especially during convective weather events characterized by rapid changes in precipitation patterns and their movement. The research focuses on three objectives: identifying the current errors and uncertainties in PySTEPS motion field estimations, constructing dynamic motion field predictions with the development of a new deep-learning model, MotioNNet, and developing motion field ensemble generation methods for MotioNNet.

We introduce a new dataset containing 10,000 motion field events, consisting of events with high rainfall and motion errors in the NL. MotioNNet, trained/validated/tested on 75%/12.5%/12.5% of the dataset, improves motion prediction accuracy by approximately 13%. It provides accurate temporal adjustments (e.g., advecting the motion), especially in structured and stable events. For ensemble generation, probabilistic techniques such as SpatialDropout and Monte Carlo dropout are employed within MotioNNet. The results indicate better ensemble predictions in structured and stable events, accurately identifying problematic areas and generating realistic members. However, it presents a small spread compared to PySTEPS. Challenges remain in accurately capturing uncertainty during highly dynamic convective events, but new paths are open for future enhancement.

The study concludes that integrating machine learning models into traditional nowcasting frameworks can substantially enhance motion field predictions and ensemble generation. Future work should focus on further refining these models or exploring more advanced architectures (e.g., ConvLSTM, diffusion models, or transformers), improving context-dependent perturbations in ensemble generation, and enhancing the references and assessments. These advancements hold potential for improving motion field nowcasting accuracy.

Acknowledgements

The development of this thesis was surprisingly and remarkably enjoyable. I became fascinated by motion fields and nowcasting, and I was fortunate to be surrounded by truly fantastic people and a supportive project environment. Therefore, I would like to express my gratitude to the following individuals whose support and help made the completion of this project possible.

First of all, I would like to thank the members of my graduation committee. Marc Schleiss, your invaluable advice, support, and the freedom granted were key to my enjoyment and progress throughout this time. Our discussions were always a powerful source of motivation. Ruben Imhoff, thank you for the enriching experience at Deltares, for introducing me to the OWM team, and for the lunches at the canteen. Our meetings at HK.3 were always an important moment of the week, where all your input, knowledge, curiosity and excitement greatly contributed to the evolution of the project. I also extend my gratitude to Riccardo Taormina for his timely ideas and engaging discussions at pivotal moments.

Furthermore, I would like to mention Sven van Os for providing me with great talks, ideas and code, especially at the beginning of the thesis. Without your help, my work would have been a much more arduous task. Peter Pavlik, thank you for sharing your expertise with NowcastNet and DelftBlue, which proved key during challenging times.

Finally, I would like to appreciate the non-technical sector of this project. Thank you to my friends, both in Spain and especially in Delft, for becoming my family away from the real one, making the good times great and the bad better. To my family for their unwavering support of my endeavours, particularly Josete and Yaya, whose help was indispensable. To Inés, I cannot stress enough how important you were for me and this project. You have been the best teammate I could have asked for (making also the best figures in this project). Lastly, to my parents, Dani and Beni, the strongest, bravest, most tender, loving and caring people I have met. You are amazing.

Daniel A. Blázquez Martín

Delft, July 2024

Contents

List of Figures

List of Tables

1 Introduction

In 2010, the World Meteorological Organization's Working Group on Nowcasting Research described nowcasting as a forecasting technique that provides detailed local weather predictions from the present moment up to six hours ahead [3]. Nowcasting plays a crucial role in various sectors that rely on weather-dependent decision-making. These sectors range from agriculture and transportation to public health and safety and early flood warning systems, impacting the security and health of millions of people [4; 5]. Given the recently documented increase in the extremity of short-lasting high-intensity (convective) precipitation events in the Netherlands [6], the relevance of Nowcasting is expected to increase in the future.

Quantitative precipitation forecasts (QPFs) have mainly been developed in recent times by using one of the two following processes: Numerical Weather Prediction (NWP) models and statistical extrapolation techniques. NWP models aim to simulate the physical processes in the atmosphere by solving a set of non-linear partial differential equations for a gridded representation of the world, parametrizing the events that occur at a smaller scale than the grid, given some initial and boundary conditions. Extrapolation-based nowcasting methods, on the other hand, predict rainfall evolution based on recent (radar) observations. These methods assume that rainfall intensity and motion fields remain constant over short periods. First, they estimate the motion field based on the previous gridded observations (e.g., radar composites) and then, by using statistical advection schemes, extrapolate the last observation along the estimated field. These nowcasts can be further refined by introducing probabilistic elements, creating an ensemble of nowcasts, therefore providing an estimation of uncertainty from the spread of the ensemble. Operational forecasting services have widely adopted extrapolation methods due to their low computational requirements and ease of implementation. Some examples of these approaches are S-PROG [7] for the deterministic case and the probabilistic STEPS [8], which have been used as benchmarks in comparison studies for new methods [9; 1]. These are implemented and publicly available in the Python-based library PySTEPS [10], a modular framework that can be considered state-of-the-art in nowcasting. This project primarily investigates the validity of the constant motion field assumption. Using a large dataset for the Netherlands, the statistical deviations between the actual and predicted motion fields are assessed and a new machine-learning based technique for generating ensemble forecasts of motion fields for different lead times is proposed.

Compared to extrapolation forecasts, NWP models are computationally intensive and produce forecasts at relatively coarse spatial and temporal scales. This limits the ability to capture convective cells, which can require km-level resolution and present lifetimes in the order of 1 hour [4]. Their update frequency and spin-up times can take multiple hours [11] which can also limit their utility in operational settings. Moreover, NWP models cannot directly assimilate radar data because radar reflectivity (or rain rate) is not one of the fundamental variables simulated, but is calculated based on more fundamental variables such as temperature, pressure or humidity. As these variables could be adjusted in many different ways to produce the same output in terms of rainfall, this can cause an assimilation problem. Despite recent improvements, such as increased spatiotemporal resolution and improved data assimilation, NWP models are still consistently being outperformed by extrapolation-based nowcasting methods for the first two hours into the future [4; 12]. After this time range, a blending of extrapolation and NWP methods shows benefits compared to extrapolation methods alone [13; 4; 14; 12].

Despite their benefits, extrapolation nowcasts also come with their own set of challenges. The presence of model errors limits the capabilities of a model, particularly the Lagrangian persistence model. As a result, the skilful lead times of nowcasting are generally limited to only a few hours (on average) into the future. These errors can arise from changes in precipitation that contradict the steady-state assumption, the assumption of a stationary motion field, the errors in the growth and dissipation of the rain field and practical implementation issues such as discretization and numerical diffusion [15; 16; 10]. Improvement in these areas can lead to better uncertainty estimation and therefore more reliable forecasts. In particular, PySTEPS has shown clear problems in forecasting extreme precipitation events related to summer convection due to the growth and decay of precipitation but also due to incorrect motion field estimation [4; 9; 1]. Growth and decay mechanisms of convective cells are often assumed to be the primary sources of uncertainty in nowcasts. Interestingly, most studies dedicated to improving the performance of extrapolation-based nowcasts mainly focus on improving rainfall intensities, while the motion field part appears to have been relatively overlooked so far. One of the first attempts to address this issue dates back to the work of [15]. Specifically, they solved a 2D system of Burgers' equation to update the motion field at each time step. This appears to be the first and only study specifically dedicated to the problem of dynamic motion field estimation. As an example, see the issue report in the PySTEPS repository[1].

Convective events usually lead to heavier, more extreme precipitation and are of great interest to the nowcasting community as they usually present the biggest challenge, and even more so for NWP models [8; 4; 14; 10; 9; 1; 12]. One of the challenges behind predicting motion fields is that they are tied to the rainfall process itself.

[1] https://github.com/pySTEPS/pysteps/issues/183

The rain cells influence the motion field, and vice-versa, which can be seen in the PySTEPS issue, where the formation of new rain cells affects the apparent motion. This is a major challenge when generating ensemble forecasts. Current methods are not dynamic enough for these types of behaviours. They only contain steady-state assumptions and random fluctuations, which are the same no matter the type of event. As such, present motion ensemble generations can be over-optimistic in capturing the true uncertainty, underestimating the errors. Another option could be using other information to include in the estimation, e.g. wind fields from NWP forecasts, but none of these alternative information sources have the spatiotemporal resolution and update frequency from weather radar. This highlights the need for further research into motion field estimation and associated uncertainties uncertainty, especially considering the high potential societal impact of such events. Probabilistic methods have been widely implemented operationally for forecasting and specifically for nowcasting. The extra information about the uncertainty in these methods is particularly useful in capturing extreme events, but it is also useful for other operational systems that could require rainfall estimate distributions and not deterministic outputs such as flood forecasting [8; 17; 9; 18]. On top of this, in the Netherlands, studies have shown that the performance of these models varies, reducing the accuracy for summer extreme precipitation events the most, mainly related to convective processes, providing only up to 30 minutes of predictability [19; 4].

Besides these more traditional methods, multiple attempts have been made to implement data-driven models using machine learning and deep learning [9; 20; 1]. These approaches are not theoretically limited for example by the steady-state assumption. They expect the model to capture the complex and dynamic evolution of rainfall fields from data alone. This has been done using different architectures (e.g. Convolutional [21; 22; 23; 24] and Recurrent Neural Networks [20] (CNNs & RNNs) or Generative Adversarial Networks [9; 1] (GANs) upon many others). Some of the downsides are 1) the low interpretability of the model, due to its black-box nature, and 2) the need for high amounts of historical datasets, which are often not present, that require high levels of pre-processing to create the cleanest possible inputs and computing power to train the model. However, the biggest appeal of these models is good predictive performance and low computational complexity without having to understand and/or explicitly model the complex physical processes. On top of that, the reduced run times, in comparison to the traditional methods, allow for the generation of larger ensemble sizes, and the possibility of customizing the models based on training data from multiple sources and from different datasets besides radar data. Therefore, a combination of the robustness and interpretability of the traditional methods combined with the speed and dynamism of data-driven models is very appealing, taking the strengths of each method while minimizing the drawbacks.

Taking the approach suggested in STEPS as a starting point, we use deep-learning techniques that take inspiration from preexisting architectures containing a motion field estimation component. With that combination, we aim to only change one component of the full nowcasting chain by predicting changes in the already estimated motion field (by PySTEPS). This hybrid approach with a focus on the motion field changes has not been seen before. As it has been described, this is one of the sources of uncertainty and research in this area can advance developments in the field of nowcasting.

1.1 Research Questions

Following from the introduction, the research questions of this MSc thesis are:

1. **What are the current errors/uncertainties associated with estimating motion fields in PySTEPS?**

   Approach: This question will be answered by comparing the predicted motion fields in PySTEPS (assuming Lagrangian persistence) and the actual motion fields for a large set of events. The motion fields are calculated from a historical dataset of radar images of the Netherlands using optical-flow methods. The average behaviour with lead time is shown by the absolute error with respect to the reference.

2. **How can the PySTEPS motion field predictions be improved using machine learning?**

   Approach: A new Deep-learning model will be trained to predict the evolution of the PySTEPS motion field. The average error calculations are repeated for the new model and the differences between the predicted and estimated motion fields will be assessed using the same methodology as in 1, comparing them to PySTEPS.

3. **How can machine learning models be used to generate ensembles of motion field predictions, and how do these ensembles compare to the ones currently implemented in PySTEPS?**

   Approach: The deep-learning model will be modified to be able to produce a prediction ensemble. The modified model and the PySTEPS ensemble generation are compared using probabilistic evaluation methods (e.g. CRPS).

   In addition to the statistical analyses, a selection of case studies will be used to illustrate the differences between the approaches and highlight the pros and cons of different types of rain events. These give insights into the spatial and temporal performance of both approaches.

2 Background

This chapter outlines the previous work that serves as the basis for this project. It begins with a detailed overview of the approach used in PySTEPS for estimating motion fields. Then, the basic Deep Learning techniques used are presented including previous usage in the field and a technical description.

2.1 Motion field estimation and modelling of errors in PySTEPS

The estimation and forecasting of motion fields in PySTEPS consists of different parts. First, the current motion field is estimated using the Lucas–Kanade (LK) optical flow algorithm [25; 26], which is implemented in the OpenCV library [27]. Other options like DARTS [28] and VET [29] are also available. Their differences are only marginal (in the order of 2% [10]) and due to their required computational times (more than 10 times compared to LK) and the increased number of required input images (2 times more than LK), the selected choice is LK. Generally, it is also the method that is most often used. LK uses two or more input images to estimate the current motion field, and the resulting field is then used in the backwards-in-time semi-Lagrangian scheme advection method [30] for the movement of the rain field, keeping the advection field constant. Once the motion field has been calculated, it is modified based on lead time by adding stochastic perturbations. The implementation is as follows:

$$\boldsymbol{w_p}(x,y,t) = \boldsymbol{w_0}(x,y) + \alpha_{par}(t)\epsilon_{par}\boldsymbol{\hat{w}_{par}}(x,y) + \alpha_{perp}(t)\epsilon_{perp}\boldsymbol{\hat{w}_{perp}}(x,y), \quad (2.1)$$

where $\boldsymbol{w_p}(x,y,t)$ is the predicted motion field at coordinates (x,y) and lead-time t, $\boldsymbol{w_0}(x,y)$ is the motion field at time 0 and coordinates (x,y), both with units of km/h, $\boldsymbol{\hat{w}_{par}}(x,y)$ and $\boldsymbol{\hat{w}_{perp}}(x,y)$ correspond to the parallel and perpendicular components of the current motion field at coordinates (x,y), being a normalized two-component vector field. The perpendicular component is defined anti-clockwise with respect to the parallel component. ϵ_{par} and ϵ_{perp} are random variables sampled from a Laplace distribution with zero mean and unit variance. They serve the purpose of introducing randomness in the perturbations. For each ensemble member, a pair of values $(\epsilon_{par}, \epsilon_{perp})$ is sampled, and for simplicity, in [10] they are assumed constant for the entirety of the field. The α-components characterize the increasing magnitude of the stochastic perturbation with lead time and, in PySTEPS, the components are modelled using a power-law model of the form:

$$\alpha_{par}(t) = a_{par}t^{b_{par}} + c_{par}, \tag{2.2}$$

$$\alpha_{perp}(t) = a_{perp}t^{b_{perp}} + c_{perp}, \tag{2.3}$$

where t is the lead time in minutes and the constants $[a, b, c]$, with units min^{-b} for a and unitless for b and c, for the parallel and perpendicular components. The constants can be chosen based on the local rainfall climatology and/or characteristics of different radar products [10]. In general, the parameters used in multiple comparison projects take the predefined standard values (see Tab 2.1). Nevertheless, these parameters can vary based on the dataset used, for example, in PySTEPS [10] parameters for other motion field estimators and radar data from Switzerland and Finland can be found (see Tab. 2.1). It is important to note that these parameters, optimized for specific events and claiming a climatological fit with a large set of fields [10], may not represent the high-intensity convective rainfall events of interest in this project. Furthermore, geographical and climatological differences, event selection, dataset size and data pre-processing (including clutter handling, interpolation, projection, discretization, corrections, and dBZ to rain-rate transformation) can influence these parameters. From [10] it is clear that the final temporal and spatial resolution of the mentioned datasets are the same, but the interpolation and other theoretical pre-processing techniques have different sources. Other differences come from the obvious climatological and geographical conditions, as well as the total grid size, for the Netherlands being [700x765] pixels compared to Finland [760×1226] and Switzerland [710×640]. Interestingly, the documentation of PySTEPS does not mention how exactly these parameters were fitted (e.g., dataset size, event selection, etc.).

These perturbations are statistically uniform across the whole spatial domain and bear no correlation with the forecast velocity, which results in perturbations not necessarily located where the most uncertainty is in the estimation. While these perturbations remain constant in space for each ensemble member, their magnitude progressively increases over time. This is done to account for the uncertainty in the prediction of the stationary motion field over time[19]. In their study, [8] showed that, when comparing an advection field at 0-lead time with that at the final lead time, there is no notable difference, both presenting very similar Root Mean Square Error (RMSE) with lead time in the nowcasts for up to 3 hours. That is why the initial motion field is used as

		LK			DARTS			VET		
		a	b	c	a	b	c	a	b	c
Standard	par	10.88	0.23	-7.68	-	-	-	-	-	-
	perp	5.76	0.31	-2.72						
Finland	par	2.21	0.34	-2.49	13.7	0.16	-16.24	0.25	0.68	11.05
	perp	2.22	0.32	-2.57	8.27	0.18	-9.54	0.02	1.00	7.40
Switzerland	par	2.56	0.33	-2.99	24.28	0.11	-27.30	0.51	0.54	7.90
	perp	1.31	0.36	-1.02	-78.08	-0.03	75.67	0.68	0.42	4.74
Fin + Swi	par	2.32	0.34	-2.65	16.55	0.14	-19.24	0.29	0.62	8.68
	perp	1.91	0.33	-2.07	14.75	0.11	-16.26	0.23	0.59	5.98

Table 2.1.: Fitted coefficients of Eqs. 2.2 and 2.3 provided in the PySTEPS GitHub repository `https://github.com/pySTEPS/pysteps/blob/master/pysteps`. It provides the parallel and perpendicular components for the standard case, the radar data of Finland, Switzerland and the combination of these two. The standard values are only shown for LK as they are the same no matter the method.

the base of the perturbations.

Looking a bit deeper into the parameters, focusing on Lucas-Kanade, the coefficients present negative intercepts (c_{par} and c_{perp} in eq. 2.2 and 2.3) with a very high correlation with each other, having a minimum value of ± 0.6 and for the a and c coefficients a perfect negative correlation (see Tab. 2.1). It can be observed more noticeably for DARTS. The high correlation present can lead to different combinations of values with similar goodness of fit. This can lead to convergence issues in the numerical methods used to derive optimal parameter values and can lead to strange, unrealistic parameter values with little or no physical meaning. This type of problem is usually known as "parameter indeterminacy" or "equifinality" [31]. In this situation, it makes it challenging to uniquely determine the values of the parameters. This can occur because of a lack of data but can also be indicative of a deeper issue, such as an inadequate or poorly constrained model for uncertainty estimation. Therefore, in these cases, it is important to consider ways to lower the visible cross-correlations, where one option could be changing the shape of the fitting equation [32]. In summary, the current ensemble generation method for the motion field poses two limitations: (1) the perturbation values are fixed in time (per event) and space (uniform), and (2) the model for uncertainty estimation may be inadequate.

It is also noteworthy to point out that during our initial investigations when we tried to fit our own parameter values on 288 events (≈ 24 h, i.e. a single day of radar images) in the Netherlands, many negative intercept parameters were found. This is disturbing from a physical point of view because a standard deviation cannot be negative, but it also creates problems for predicting the error variance over short lead times and generating realistic ensembles. Moreover, the fitting generated unexpected extremely

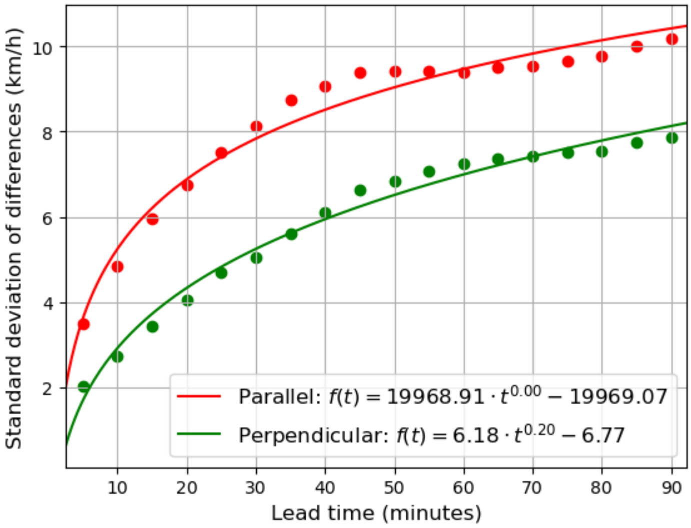


Figure 2.1.: Parameter fitting output for 24 hours in the Netherlands on 28/07/2014. For the parallel component, the fitted values explode in comparison to the expected values, in the order of magnitude of the perpendicular case.

high and low values that compensate for each other (see Figure 2.1). This shows that the selected model presents clear weaknesses in its definition and constraints. While this may not be a big issue from a practical point of view since usually, the standard deviation quickly grows and becomes positive after about 5 minutes, it is a serious methodological weakness in the current error model. In the end, the current approach used by PySTEPS is functional but lacks physical interpretability and scalability to other types of data with different lead times.

2.2 Deep Learning Architecture: U-Net

The U-Net architecture was introduced for image segmentation with great success in [33] and then has been applied to many other fields, including nowcasting [21; 22; 23; 24] due to its advanced feature extraction and processing capabilities from image inputs. The architecture structure is based on an encoding-decoding strategy heavily dependent on convolution and pooling with skip connections to preserve spatial information. Over time, the vanilla implementation of the U-Net has been refined and in particular this project focuses on the approaches followed in [9; 1]. These present modified U-Nets with changes in the architecture and loss adjustments to improve previous existing U-Net-based models. Convolutional layers are substituted with residual blocks and rather than predicting only a single output and using auto-regressive sampling, their models predict all frames in a single forward pass. On top of that, they include a loss weighted by precipitation intensity. All these modifications are introduced to reduce blurring and improve performance. The results show still an important blurring effect which produces an approximately 8 km effective resolution at 30 min lead time and 32 km at 90 min in rainfall nowcasting [9]. Even though these properties are not desired in an end-to-end prediction (and therefore penalize U-Net with respect to other approaches), these are usually regarded as good or desired characteristics of motion fields [8; 15], as motion fields tend to be smooth over longer distances, making U-Nets a good contender for motion field estimation.

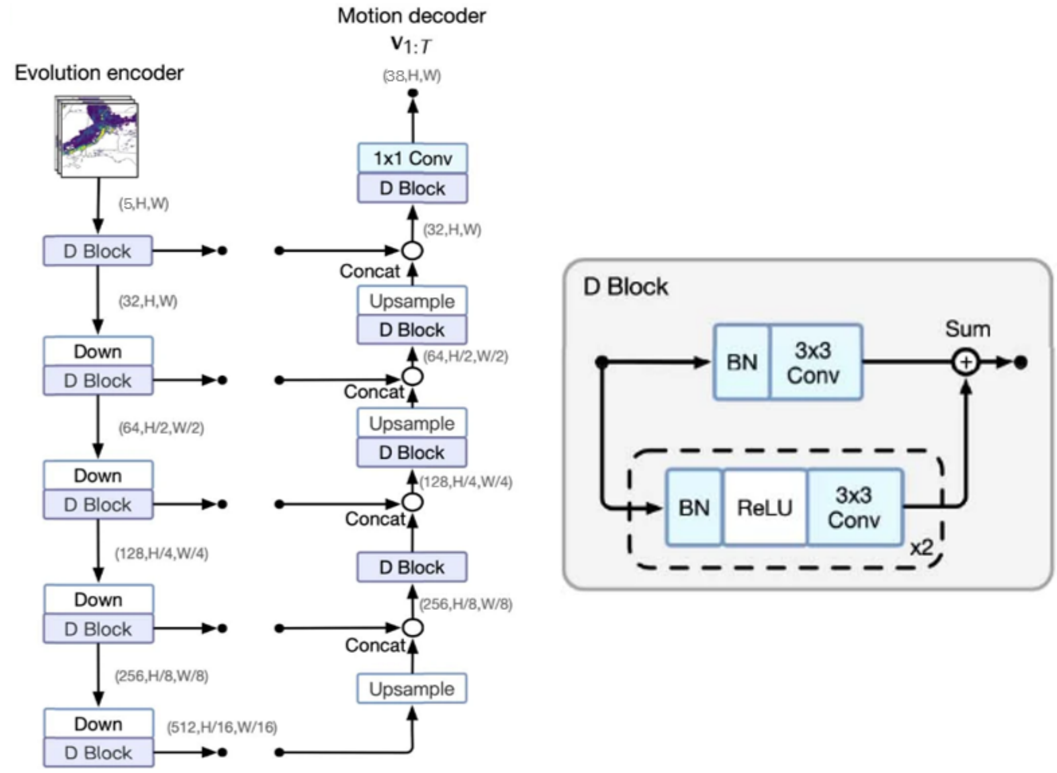


Figure 2.2.: Modified figure from [1]. Architecture details of the elements in the Evolution Network (EvN) used for this project. It shows the EvN encoder with specifics on the D Block. The arrows leaving the D-Blocks and posterior concatenations are the skip connections. The input fields have a 480x480 shape in this project, with convolutional layers using a 3x3 kernel. Leaky ReLU is used as activation function with a negative slope of 0.2. BN is the batch normalization. Up and Down are bilinear interpolations to expand or reduce spatial size. All the predictions are done in a single forward pass.

Nowcastnet's Evolution Network (EvN) [1] is the closest approach to the one of interest here, as it explicitly predicts motion fields. The EvN has a shared encoder for latent representations, a motion decoder for learning future motion fields and an intensity decoder for the growth and decay of precipitation. These predictions are then combined using a backward semi-Lagrangian scheme to advect the rainfall field and the model is trained to minimize the loss between the advected and observed fields together with a continuity regularization applied to the motion field. Some of the physical concepts that are used in the background of this model also correspond to what is used in PySTEPS, which makes this approach more appealing for this project. As this model aims to create an end-to-end prediction, the sections related to advection, growth and decay are skipped and only the motion prediction approach is extracted. Specific details on the architecture can be found in Figure 2.2. It consists of five residual blocks (D-Blocks), where each block doubles/halves the number of channels followed by spatial down/up sampling by a factor of two. The number of channels ranges from 32 to 512 in the latent representation. The implementation and modifications made to this model are further described in Section 3.5.

2.3 Probabilistic Deep Learning: Dropout

Dropout [34; 35] is a widely used regularization technique introduced to prevent overfitting in neural networks. This occurs when a model learns the training data too well, capturing noise along with the underlying pattern, and hence performs poorly on unseen data. The standard dropout implementation operates by randomly setting the activation for a fraction of the neurons to zero (they are “dropped-out”) at each update during training with a certain probability. The core idea behind dropout is that it introduces noise into the training process, forcing the neural network to learn more robust and generalized features.

While dropout has proven to be an effective regularization technique, it has certain limitations when applied to Convolutional Neural Networks (CNNs). In CNNs, dropout is less effective due to the strong spatial correlation between adjacent pixels in images [36; 2]. Simply dropping out pixels does not fully prevent neurons from training. Spatial Dropout [2] is a variant of the standard dropout technique, designed specifically for convolutional layers (see Fig. 2.3). It disconnects entire feature maps in a 2D convolutional layer, which effectively prevents the learning of the kernels. This method preserves the strong spatial correlation between adjacent pixels in feature maps, as the maps not dropped out have the entire domain available for training, while still providing a regularizing effect.

Monte Carlo Dropout, an extension of the dropout method, allows us to obtain a measure of uncertainty in deep learning models. By performing dropout at test time, we can generate an ensemble of predictions. The variance of these predictions can then be used as a measure of uncertainty while also having multiple different predictions. This method has been shown to improve the performance of CNNs and U-Nets by providing a more robust prediction. The implementation of the Dropout method to construct ensembles is further described in Section 3.6.1.

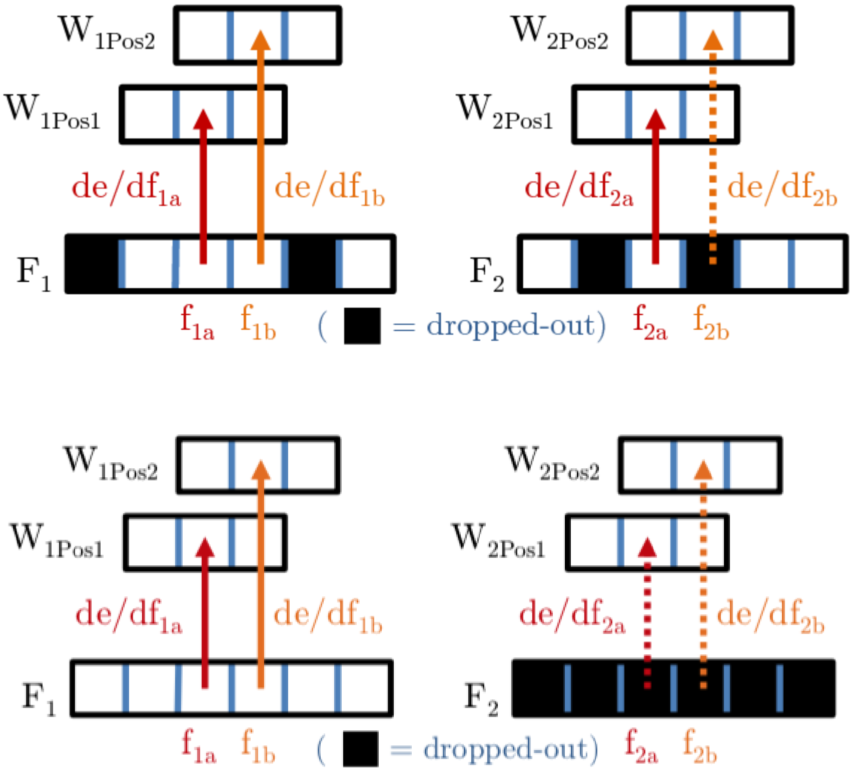


Figure 2.3.: Modified figures from [2]. Dropout vs SpatialDropout. The figure illustrates the convolution process with kernels represented by the top two pixel rows for feature maps 1 and 2, while the bottom row depicts the output features of the preceding layer. At the top, during back-propagation, the W2 kernel's centre pixel receives gradient contributions from both f2a and f2b, which is dropped, as it moves over the input feature F2. As these are highly correlated pixels, despite the zero gradient contribution to minimizing e (the loss function) from f2b, the strongly correlated f2a gradient persists and independence is not enhanced. On the right, with SpatialDropout adjacent pixels in the dropped-out feature map are either all 0 or all active, therefore enhancing the missing independence in the standard dropout procedure.

2.4 Probabilistic Evaluation: CRPS

When generating multiple predictions, probabilistic evaluation methods need to be applied to account for the quality of the ensemble and not just compare the errors of the different members independently. A very common verification tool used for probabilistic prediction is the Continuous Ranked Probability Score (CRPS) [37; 38; 39]. Based on the description in [39], it is defined as follows:

$$\mathrm{CRPS}(P, x_a) = \int_{-\infty}^{+\infty} [P(x) - P_a(x)]^2 dx \tag{2.4}$$

$$P(x) = \int_{-\infty}^{x} \rho(y) dy \tag{2.5}$$

$$P_a(x) = H(x - x_a) \tag{2.6}$$

$$H(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases} \tag{2.7}$$

where $\rho(x)$ is the Probability Density Function (PDF) given by an ensemble forecast and x_a the observation. $P(x)$ and $P_a(x)$ are cumulative distributions and $H(x)$ is the Heaviside function. The CRPS measures the difference between the predicted and occurred cumulative distributions. The idea behind this tool is to combine the average error of the members with their spread value. It is zero when P and P_a are equal (all the members are exactly the observation) and has the dimension of the parameter. If the prediction is deterministic the CRPS reduces to MAE making it easy to interpret. The mentioned description focuses on continuous distributions but for the discrete cases the following modifications need to be made:

$$P(x) = p_i = \frac{i}{N} \text{ for } x_i < x < x_{i+1} \tag{2.8}$$

where $P(x)$ is now a piecewise function, x_i is the ith ordered outcome of the ensemble where the first and last elements are set to $-\infty$ and $+\infty$ respectively for convenience. This approach assumes equal weight for each of the members in the prediction. The implementation of CRPS for this project is described in Section 3.6.2.

3 Data and Methods

This chapter describes the specific tools, approaches and information employed in this thesis to manipulate, filter, estimate and asses motion fields needed to reproduce the work.

3.1 Radar data

Motion fields are calculated using the 5-minute KNMI radar reflectivity composite maps. This product has been archived since 2008 and contains gridded files, with dimensions 700x765, at on average 1500 m altitude over the Netherlands and surrounding areas measured by two radars in Herwijnen and Den Helder and can be found in https://dataplatform.knmi.nl/dataset/radar-tar-refl-composites-1-0. A pre-processing step has been implemented on this dataset to remove clutter using a morphological clutter removal scheme by [40] in Section 4.1. The radar data provided by KNMI are stored in an 8-bit unsigned integer representation corresponding to a range from 0 to 255. It first needs to be converted to reflectivity expressed on a logarithmic scale in units dBZ. The transformation to dBZ is done using the following equation:

$$\begin{cases} Z(x_i, y_j, t) = \text{NaN} & \text{if } K(x_i, y_j, t) = 255 \\ Z(x_i, y_j, t) = K(x_i, y_j, t) \cdot 0.5 - 31.5 & \text{otherwise,} \end{cases} \tag{3.1}$$

where Z and K are the radar reflectivity in dBZ and bit-wise representation respectively of the pixel at coordinates (x_i, y_j) and time t. If needed, the reflectivity values can also be transformed to rain rates (based on the Z-R Marshall-Palmer [41] relationship with a=200 and b=1.6):

$$R(x_i, y_j, t) = 10^{(Z(x_i, y_j, t)-23)/16}, \tag{3.2}$$

where $R(x_i, y_j, t)$ is the rain intensity in mm h^{-1} of the grid cell at coordinates (x_i, y_j) and time t.

3.2 Event Selection

The selection is derived from the approach proposed in Section 4.2 of [40], where an event is defined as a sequence of 22 consecutive radar images, with 4 input frames and 18 prediction frames, separated by 5-minute time intervals, focusing on a period of 90 minutes. For an event to be considered “valid”, both KNMI radars must be recording data. Additionally, all the radar reflectivity and echo top height maps for the 22 consecutive frames must be available. For each valid event, a weighted sum of the rainfall intensities at the pixel level is computed:

$$S_k = \sum_{t=1}^{22} \sum_{ij} R(x_i, y_j, t)^p, \tag{3.3}$$

where S_k indicates the calculated coefficient of event k, t corresponds to the time index of each frame, $R(x_i, y_j, t)$ is the rain intensity in mm h^{-1} of the pixel at coordinates (x_i, y_j). The indices i and j correspond to the position of the pixels, in the x and y direction, for a central crop of [480x480] from the original [700x765] (taking indices 109 to 589 for x and 141 to 621 for y) domain. The analysis is restricted to this area to mitigate larger errors caused by border artefacts in radar composites, while fully covering the Dutch land mass. The restricted area is kept for the entirety of the project. p is a unitless exponent that can be used to change the weight given to individual pixels as a function of their intensity. For example, if $p = 1$ all rainfall intensities are treated equally but if $p = 2$ more importance is set to pixels with higher intensity, focusing on events that are more localized but more extreme rather than more widespread with less severe rain. In this project, $p = 2$ is chosen to emphasize highly convective and localized events.

For this project, a “valid” event must consist of a sequence of 25 consecutive frames (as extra rain frames are needed to calculate the motion fields, see Sec. 3.3), compared to the 22 of S_k. The restrictions on the availability of the data are preserved. Even though the difference in length of a “valid” event can create conflicts (where there are 22 available frames but not 25, e.g. one radar stops recording), the impact caused by this difference in the coefficient S_k is small, losing a fraction of “valid” events. Therefore, the selection of events is kept the same considering only the first 22 frames. Out of the desired amount of events, 30,000 in this study, the 3-frame difference leaves us with 27,740 events available.

The mentioned procedure can generate a dataset with greatly correlated and overlapping events, which can significantly imbalance the distribution of events, over-representing single events with high intensity and long duration. This impacts the error analysis and the training of the machine-learning models through overfitting or training slowdown. To avoid this, a multiple-step approach is followed for the selection. This is done sequentially, using a list of sorted events based on the associated value from eq. 3.3.

First, the sequence of 22/25 radar observations (the event) with the highest value is chosen and then all the events inside a 3-hour range around it, as declustering time, get removed. The declustering time is introduced to guarantee a certain level of independence and is based on [8], which shows that after this period events start being substantially different. This process is repeated, selecting the next highest value of the remaining events in the list and removing the surrounding time steps. This is done until the final amount of events is reached.

3.3 Motion Field Estimation for Training and Reference

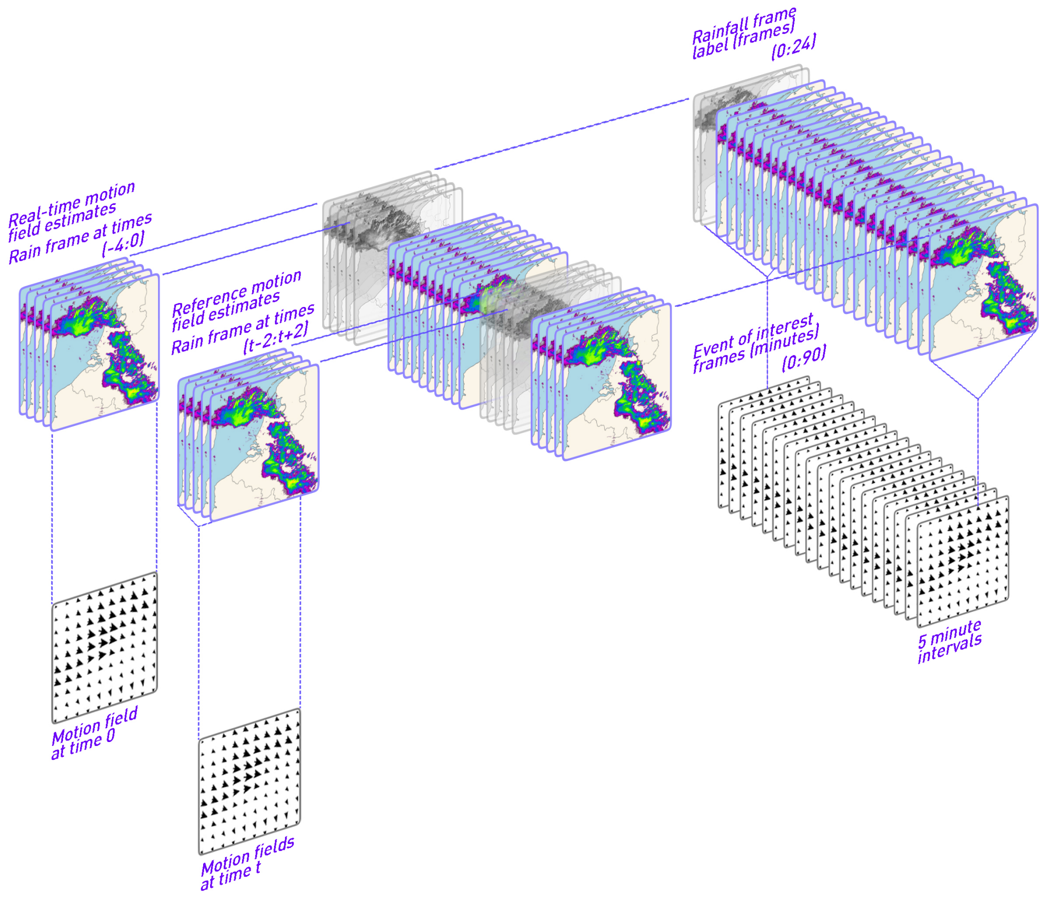


Figure 3.1.: Diagram of the motion field calculation procedure. PySTEPS dataset is composed of only one motion field that uses the first five rainfall images. The reference dataset computes a motion field for every time step in the event of interest using a window of 5 frames around the current time t for 90 minutes. The naming convention uses the time label of the first rainfall frame.

This section presents the approaches followed to calculate the motion fields from the datasets used for training and as reference. Only specific modules from PySTEPS (v1.7.4) were used in these calculations, including the data importers (for the KNMI radar data), the Lucas-Kanade optical flow calculation module and transformation modules to pre-process the radar rainfall data to scaled values (see Sec. 3.1), which is needed for the Lucas-Kanade module [26; 25; 10].

This project focuses on events with a fixed duration of 90 minutes (19 frames). In practical terms, this means that for each event, we need to extract 25 frames (19 + 4 extra input frames + 2 extra frames at the end to compute the reference motion field for the 19th timestep in the event). The naming convention is based on the time of the first file, out of the 25. The output motion fields are given in km/h and are always calculated using 5 radar composites in dBZ units. Two different approaches are implemented:

1. U_{rt}: online calculations. Motion fields are computed in real-time, using only data from the past and present. It considers one single motion field per event and is computed using the first five images (see Fig. 3.1). These files correspond to the frame where the prediction begins and its preceding four frames. This is the approach followed in PySTEPS.

2. U_{ref}: offline calculations. Motion fields are computed using past (previous 2 frames), present and future (next two frames) radar data. It uses "future" information that is not available for U_{rt}. The calculations cover the following 90 minutes into the future, therefore creating 19 motion fields. The first motion field is centred at the start of the prediction. This gives the best possible estimate for every lead time a-posteriori. This approach is considered as the reference (see Fig. 3.1).

3.4 Selection of Interesting Events

To remove events that contain a lot of clutter or insignificant rain, which could also impact the motion estimation process, an empirical coefficient and cap were introduced. After the implementation of the declustering time, introduced in the previous section, we removed events without rain or filtered by the cap. This, together with focusing on the events with the largest coefficient, reduced the initial 30,000 events to 10,000. This empirical method is based on preliminary analyses. It seems to work well in practice, meaning that it eliminate problematic events and extracts interesting events (based on visual inspection), but it is not a critical aspect of the work and could be modified.

The coefficient tries to combine the velocity error and the presence of rain. It is calculated as follows:

$$V_k = (\ln{(S_k + 1)})^2 \cdot \ln{(\sqrt{\overline{\Delta V_x^k}^2 + \overline{\Delta V_y^k}^2} + 1)}, \tag{3.4}$$

$$\overline{\Delta V_c^k} = \sum_{x_i, y_j, t} \mid U_{ref,c}(x_i, y_j, t) - U_{rt,c}(x_i, y_j, t) \mid \cdot \frac{R^2(x_i, y_j, t)}{S_k}, \tag{3.5}$$

where S_k is the coefficient from equation 3.3, $\overline{\Delta V_x^k}$ and $\overline{\Delta V_y^k}$ are the averaged in time and space velocity rain weighted absolute errors of the x and y components, calculated using equation 3.5, where $\overline{\Delta V_c^k}$ is the average error of component c. The rain weights are computed by dividing each pixel's rain intensity in mm h^{-1} to the power of two ($R^2(x_i, y_j, t)$) with S_k. These weights multiply the error in the velocity and the sum of all of them provides a spatial and time average error. The natural log is introduced as a form of normalization, and the selection of $\ln x + 1$ ensures positive values for the coefficient.

The idea behind V_k is to give more importance to events with lots of rain, high intensities and big errors in velocity estimates while trying to avoid clutter and artefacts, which can impact large areas of the motion field calculations (see Ap. A.3). The focus on rain and velocity errors is crucial. The first serves as a way of reducing the effect of clutter and artefacts, as the impact on large areas of motion gets minimized when large rain fields are present and balancing the impact and relevance of the event. The second is at the core of the research interests of this project, trying to improve in the areas where the static approach does not work well.

Despite implementing measures to reduce clutter and artefacts, which highly influence the motion field estimation, certain events still exhibited significant deviations. To address this, a cap on the velocity error is considered at 35 km/h, which means that the domain-wide average velocity error has the magnitude of approximately the average tropical depressions [42]. If any event at any lead time contains a spatially averaged error in any of the motion components above the limit, it gets removed. There are examples where the error can go up to 200 km/h making this a key problem to tackle. The value selection is based on the magnitude of events of interest in comparison to outliers, and taking into account typical wind speed velocities. For context, the 35 km/h cap corresponds to the 99.2th percentile of the velocity errors in events with rain at a 90-minute lead time (see Fig. B.1). This procedure removes approximately 3% of the events with rain (see Ap. A.3).

3.5 Deep Learning Methods for Dynamic Predictions

3.5.1 Motion Field Prediction with MotioNNet

Based on the work described in Sec. 2.2, substantial modifications to EvN have been implemented to enhance and adapt the model based on the research purposes.

The basic architecture can only extract information from 5 rainfall frames in order to predict the motion fields 90 minutes into the future. This is a challenging task especially when interpolating the motion to areas with no rain and when changes are large and highly dynamic in time. To overcome this problem, architectural and loss changes are implemented. The most important change concerns the input of the model. In addition to using the 5 most recent radar frames, the new model, from now on called MotioNNet (MNN), also takes the real-time estimated motion field U_{rt} from PySTEPS as an input (see Fig. 3.2). Then, instead of predicting the motion field directly, MNN is modified such that it predicts the changes to be made to U_{rt}, for the next 90 min. in a single forward pass, to get as close as possible to the reference motion field U_{ref}. These two implementations are meant to benefit from the current PySTEPS approach and then use Deep Learning to increase its flexibility and dynamism in the prediction. The architecture and loss details are as follows:

1. As past motion is given as input, the architecture now contains two encoders (see Fig. 3.2), one for rainfall and the other for motion, which extract the relevant information from both sources. Then, the latent representations are concatenated and processed through a D-Block (see Fig. 2.2) to generate a combined representation that is fed into the decoder. The skip connections only apply to the motion fields, as the fine details of the rainfall fields in space are not relevant to be preserved. The extra information introduced by the past motion provides a robust baseline for the model. It is more noticeable in the areas that never present rain, where otherwise the model would not have any input information to predict the motion. This allows the model to focus better on what needs to be changed in the field and not predict the entire motion from scratch.

2. Giving the motion as input can also cause issues. When the model tries to predict future fields, it does not make substantial changes in time when required, going back to U_{rt} and remaining static. To tackle this, the model does not target U_{ref} but the difference between U_{ref} and U_{rt}. This, when added to U_{rt}, produces a new set of motion fields. That way, the model is forced to be more accurate in the changes of the motion and can remain static in the areas where no changes are expected to happen, keeping the interpolation capabilities of the PySTEPS approach.

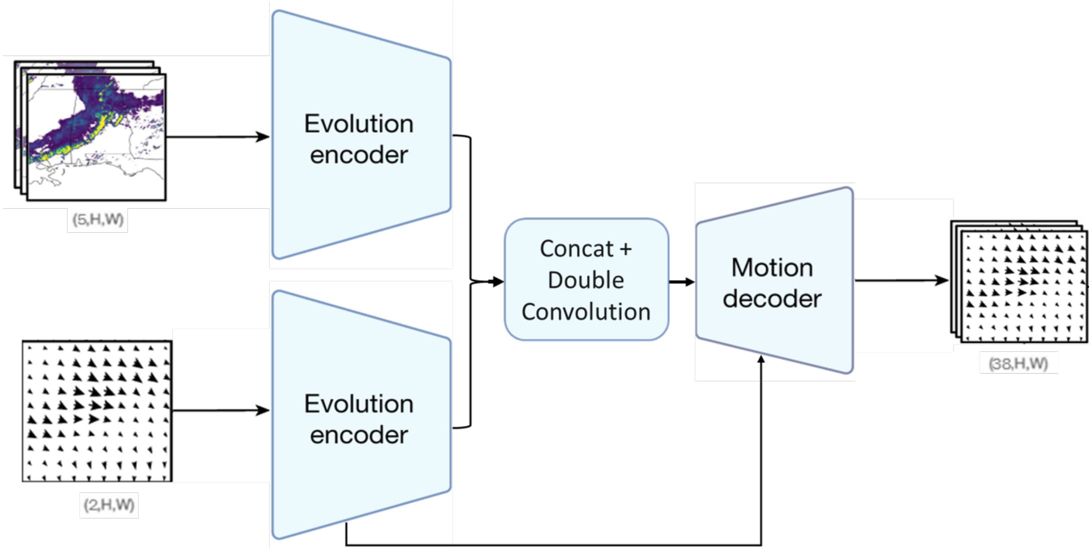


Figure 3.2.: Modified figure from [1]. MotioNNet (MNN) Architecture details based on Sec. 2.2. The diagram displays two evolution encoders, one for the rain inputs and the other for U_{rt}. After encoding, the results in the latent space get concatenated and double convoluted to combine the learned representations. Then the results go through the motion decoder to output the predicted error, in a single forward pass, of U_{rt} compared to U_{ref}. The skip connections only happen between the U_{rt} encoder and the decoder.

3.5.2 Training, Validation and Testing

The model loss is based on the Mean Absolute Error (MAE) defined as $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$, where y_i is the target, $\hat{y}_i$ the prediction and n the number of pixels. The MAE is calculated between the prediction and the difference of U_{ref} and U_{rt}. The absolute error is calculated per grid cell and then averaged over time and space. The model was trained on 20,000 iterations with a batch size of 16, a learning rate of 0.001 and a decreased learning rate on plateau of 0.1. The hyper-parameter tuning and training of the model was done completely on the DelftBlue Supercomputer from TU Delft [43], using NVidia Tesla V100s 32 GB GPUs. The dataset is divided randomly into training, validation and testing using 75/12.5/12.5 % respectively (7500 events for training, 1250 for validation and 1250 for testing). The sizes and content of each group showed in the training to be sufficient to represent the diversity of events in the dataset.

In addition to the statistical analyses performed on the testing dataset, we also focus on some hand-selected interesting case studies which will be used to illustrate the pros and cons of the different approaches. If these events were present in the training or validation set, they were moved to the test set. The selected events try to highlight the

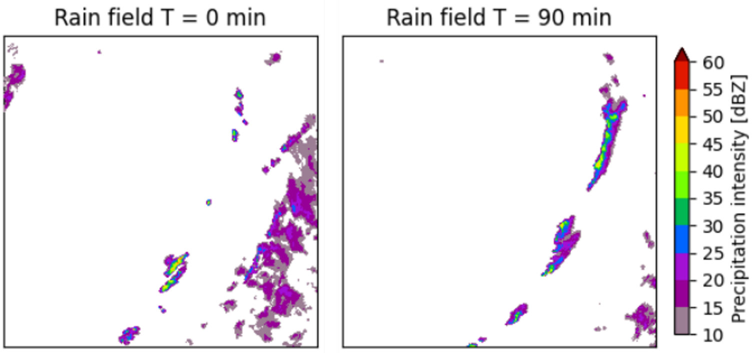


Figure 3.3.: Rain evolution of the squall line event from 2012-06-16-1755. It presents the rain frames at time 0 and 90 min. in dBZ.

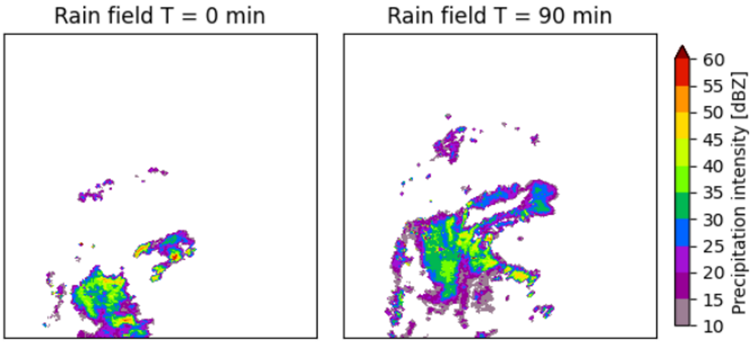


Figure 3.4.: Rain evolution of the isolated convective cells event from 2010-07-10-1800. It presents the rain frames at time 0 and 90 min. in dBZ.

magnitude of the errors that can take place due to the static motion field assumption. The events are 2012-06-16-1755, 2012-01-03-1410, 2021-10-21-1025 and 2010-07-10-1800. They represent two of the main characteristics found when exploring cases using V_k. The first three events are associated with highly structured and stable events containing what is known as squall lines (see Fig. 3.3 and C.1) and cyclonic rotation (see Fig. C.4) and the last related to isolated convective cells (see Fig. 3.4). Of the first three events, only the squall line of 2012-06-16-1755 will be shown from now on as it summarizes the most important features of these types of events, but the analysis for the other two can be found in Appendix C.

Squall lines are quasi-linear convective systems where convection is heavily structured following a line. They tend to have a steady structure and fast displacement, lasting several hours [44]. These types of events have been found multiple times in the dataset of interest, and the structured line of convection can be fully connected or disconnected, where the latter presents higher-intensity precipitation. Squall lines are of great interest as the impact of the motion field estimation can be comparable in magnitude to the growth and decay of precipitation, which is common to other highly structured and stable events. In these events, the motion field gets advected following the squall line, as it does not fully follow the general motion direction, but can deviate in movement from this. This property is shared for convective areas in other structured and stable events (see Ap. C). That way, as the convective line propagates, the motion field propagates with it, avoiding the intrusion of the rain in areas with much different displacement. Preserving the motion of the line is important, as squall line motion tends to be predominant when encountering other rainfall structures. In the squall line event selected (see Fig. 3.3), there is a disconnected line of convection moving eastward. It pushes a region of lighter widespread rain to the east which itself moves very quickly NE.

Isolated convective cells, on the other hand, are much more intricate in the relationship between motion and growth and decay. These cases usually present one or more isolated convective cells that are independent of the general movement and when multiple isolated cells are present, more randomly spread. This imposes a great challenge in the nowcast, as the motion field evolution is strongly connected to the growth and dissipation of the cells, requiring the estimation of the motion field and the estimation of the rain evolution as well. In the selected event (see Fig. 3.4), the entire domain moves northwards. Isolated cells with high-intensity precipitation in the centre of the area split and connect with the bigger area of rain to the left, dissipating slightly. On top of that, more cells get generated connected to the bigger area of rain at the right.

3.6 Ensemble Prediction

3.6.1 Ensemble Generation

The techniques described in Section 2.3 are applied to MNN to compare the ensemble generation with PySTEPS (see Sec. 2.1). All the models involved in the comparison generate ensembles with 30 members.

To each D-Block (see Fig. 2.2) inside MNN (see Sec. 3.5), SpatialDropout was added by dropping feature layers in the output of both the double convolution and the single convolution separately. The model was trained for different drop probabilities: 0.1, 0.2 and 0.5. This range provides information on the effect dropout has on the model. For higher probabilities, worse performance per member but greater diversity and spread are expected. The SpatialDropout was kept during testing to generate an ensemble of predictions following the Monte Carlo dropout technique.

For a fair comparison with the PySTEPS approach, the coefficients described in Sec. 2.1 are fitted to the 10,000 selected events. The resulting values are [0.4826, 0.5241,

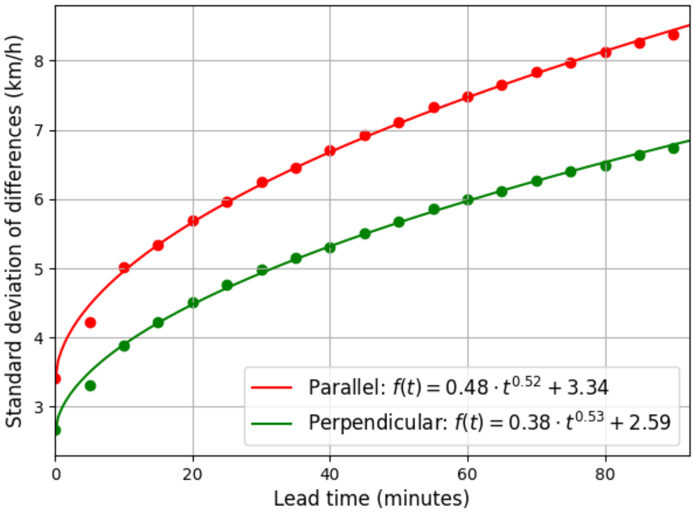


Figure 3.5.: Standard deviation of the differences in the motion field dataset between U_{ref} and U_{rt}. The fitted parameters of Eqs. 2.2 and 2.3 are included.

3.3406] for the parallel component and [0.3813, 0.5328, 2.5928] for the perpendicular component (see Fig. 3.5 and Eqs. 2.2 and 2.3).

3.6.2 Evaluation of the ensembles

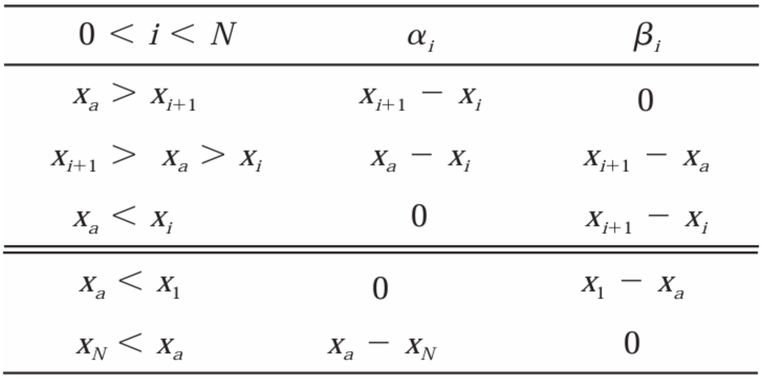
For the evaluation of the PySTEPS and MNN with dropout ensemble generation approaches, we use the CRPS coefficient described in Section 2.4. The metric, with km/h units, is calculated for the entire domain ([480x480]) for all lead times. It can then be averaged in space to visualize the temporal evolution of the models. The performance over the spatial domain can be shown for a single lead time including and comparing individual members.

In particular, the approach shown in [39] and implemented in PySTEPS is used. This implementation is needed due to the finite size of the ensemble, which requires additional steps from Section 2.4. The CRPS in this approach is evaluated as follows (see Sec. 2.4 for notation):

$$c_i = \int_{x_i}^{x_i+1} [p_i - H(x - x_a)]^2 dx \rightarrow \text{CRPS} = \sum_{i=0}^{N} c_i \tag{3.6}$$

and c_i can be evaluated based on the location of the observation as:

$$c_i = \alpha_i p_i^2 + \beta_i (1 - p_i)^2 \tag{3.7}$$



$0 < i < N$	α_i	β_i
$x_a > x_{i+1}$	$x_{i+1} - x_i$	0
$x_{i+1} > x_a > x_i$	$x_a - x_i$	$x_{i+1} - x_a$
$x_a < x_i$	0	$x_{i+1} - x_i$
$x_a < x_1$	0	$x_1 - x_a$
$x_N < x_a$	$x_a - x_N$	0

(3.8)

where α and β have km/h dimensions. These equations provide the CRPS per pixel. As all the pixels and members are considered to have the same weight, the standard arithmetic mean is used to obtain spatial or temporal averages.

4 Results

4.1 Error Analysis of Motion Field Estimation

4.1.1 PySTEPS Statistical Characterization

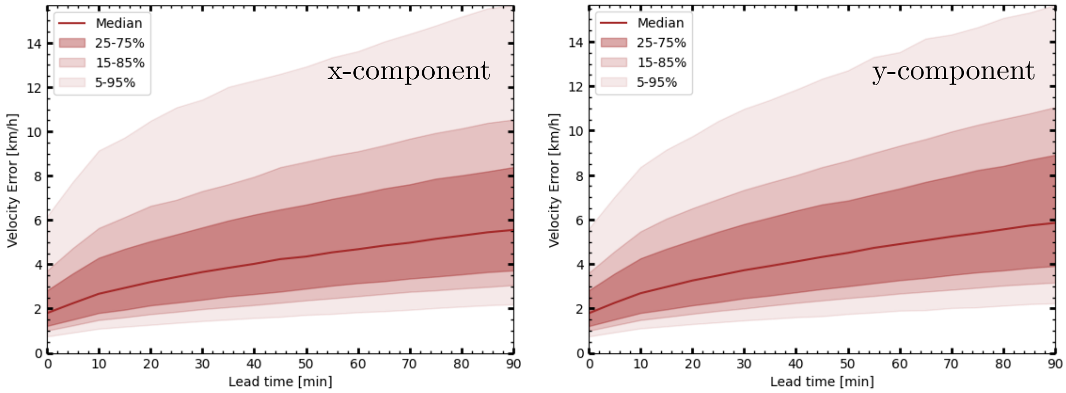


Figure 4.1.: MAE of U_{rt} vs U_{ref} for the top 10,000 events selected including both the x (left) and y (right) components. Different percentiles are shown indicating the spread of the error in the different events. The associated displacement error has also been calculated and can be found in the Appendix B.

Following the methodology described in Sections 3.3-3.4, we calculated the spatially averaged MAE of the estimated motion fields for the x and y components. Figure 4.1 shows that the PySTEPS approach already deviates at the first time step, indicating that using only past observations for the motion field is not enough. There is a rapid and sudden increase in MAE for short lead times below 10 minutes. The median value over time is approximately 4-9 km/h for both the x- and y-components, leading to an overall misplacement in the order of 3-8 km in 90 minutes. These magnitudes refer to an average over the entire domain, but locally, the errors can be substantially larger. The distribution of the errors is positively skewed for the entire range with an increasing spread at longer lead times, magnifying the errors in the velocity estimation (up to 15 km/h at 90 min for the 95th percentile) and therefore the displacement. The comparison with the results including all the events with rain (without filtering down to 10,000 events) can be found in Appendix B.

4.1.2 PySTEPS Analysis on Events of Interest

The analysis of the events of interest (see Sec. 3.5.2) shows that U_{rt} lacks a more dynamical prediction of the motion fields, which causes significant errors in the motion and therefore large misplacement issues. For highly structured and stable events like squall lines, these errors are very prominent (Fig. 4.2). When looking into isolated convective cell events, the errors are again present, but growth and decay of precipitation play a bigger role, which does not allow us to solve the issue by simply advecting the motion (Fig. 4.3).

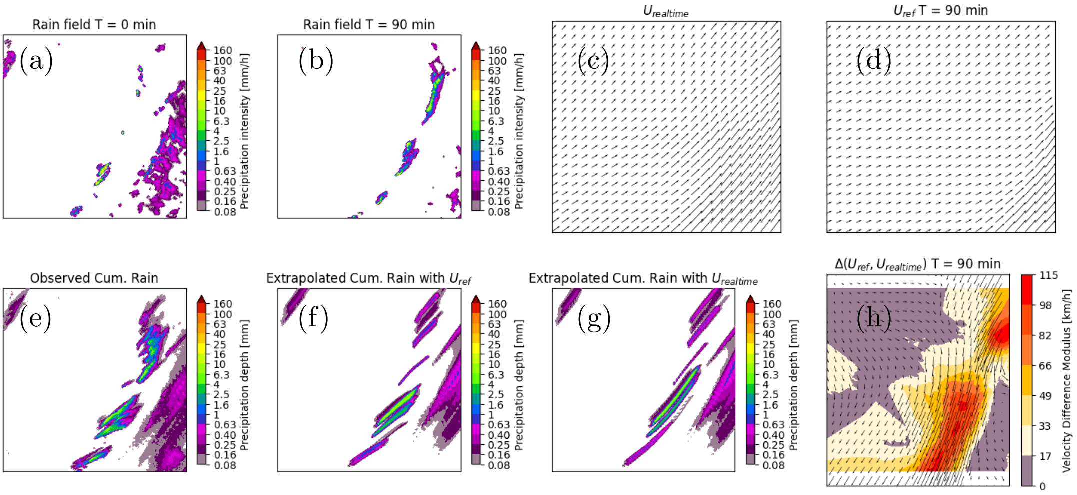


Figure 4.2.: Representation of the squall line event on 2012-06-16-1755. It contains the rain observation at 0 and 90 min. (a, b), U_{rt}, U_{ref} at 90 min. (c, d), their difference at 90 min (h). and the observed and extrapolated cumulative rainfall over 90 min. using the static and the dynamic approach for the latter (e, f, g).

In the squall line event (see Fig. 4.2), there is a clear identification of a problem in the region over which the squall line moves. As U_{rt} does not advect the motion field, the errors grow up to more than 100 km/h. This can significantly affect the nowcast of rain. For example, the results of using U_{ref} for the extrapolation of rain shows that the easterly movement and the structure of the rain are better preserved than when using U_{rt}(see Fig. 4.2). There are clear changes due to growth and decay, but the spatial structure of the convective cell at the SE of the centre with intense precipitation is well captured. In the U_{rt} case the cell enters the area with high displacement and wrongly gets significantly displaced NE, together with the surrounding rainfall fields that are not part of the squall line. It is important to note that there are still modifications related to the growth and decay of rain that are not taken into account in the nowcasts, but the more accurate representation of the final fields is a positive signal towards improvement.

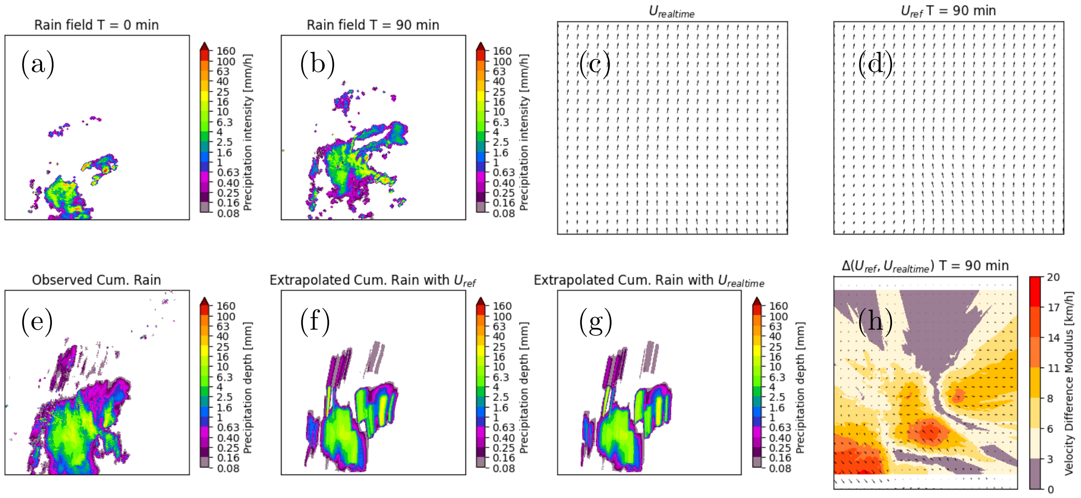


Figure 4.3.: Representation of the isolated convective cells event on 2010-07-10-1800 related to isolated convective cells. It contains the rain observation at 0 and 90 min. (a, b), U_{rt}, U_{ref} at 90 min. (c, d), their difference at 90 min (h). and the observed and extrapolated cumulative rainfall over 90 min. using the static and the dynamic approach for the latter (e, f, g).

For the isolated convective cells events, as can be seen in Fig. 4.3, U_{ref} tries to replicate the movement of the rain. See the slopes and structure of the lines of cumulative rain at the N/NE, which are more similar to the observations. Unfortunately, as the evolution of the rainfall field is not included, the final structure is erratic. The correction produced by implementing U_{ref} makes a slight improvement when displacing the biggest two cells, but this is limited again by the feedback between rain and motion. The areas of motion that change from U_{rt} to U_{ref} are directly related to areas where rain is present, and the magnitude is bigger in the regions where growth and decay of precipitation are more intense.

From the events of interest, it is clear that a more dynamic approach in PySTEPS is desired. For the structured and stable cases, we consider U_{ref} an interesting target as it portrays correctly, in general, the evolution of the motion field. For the cases with isolated convection and more impact of growth and decay, U_{ref} help is useful but less significant. A combination of the issues of motion together with the growth and decay of precipitation is more relevant. However, in some cases, the rain surrounding the convective areas with a different motion than the convective cell can confuse U_{ref} calculations (see C.1.1). This shows some of the limitations in the reference used for this project.

4.2 Motion Field Estimation with Deep Learning

4.2.1 Deterministic Comparison of Motion Estimation

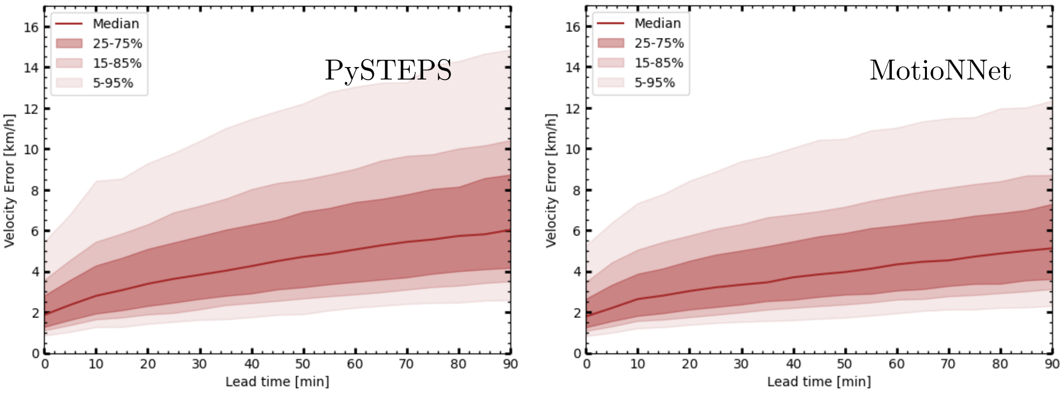


Figure 4.4.: MAE comparing the errors with lead time of U_{rt} (left) and MotioNNet (right) for the test set of the model training. It includes the average of the x and y components showing different percentiles that indicate the spread of the error.

Figure 4.4 shows the MAE of the estimated velocity by PySTEPS and the MNN, following the same approach as in Section 4.1.1. MNN provides an overall improvement (average error of 4.5 km/h) compared to U_{rt} (average error of 5.2 km/h) of around 13% for all lead times. For shorter lead times there is no significant improvement, preserving the initial error and the sudden and rapid increase observed in Sec. 4.1.1 (see Fig. 4.4). As lead time increases, the benefit of the new model is more prominent. Focusing on 90 min. lead time for ease of interpretation, the spread is reduced by approximately 18% and the entire distribution is moved downwards. The larger error events get more affected, with an improvement going up to 2.5 km/h (20%) for the 95th percentile. This demonstrates a clear impact of the new approach on the most severe cases, flattening the distribution of the errors and decreasing the spread with lead time significantly.

4.2.2 Motion Field Estimation on Events of Interest

The events of interest show that MNN performs significantly better than PySTEPS in the prediction of the motion field for highly stable and structured events (Fig. 4.5). When looking into the isolated convective cells events, where growth and decay is much more relevant, the model performs almost identically to PySTEPS while making changes to the given motion field (Fig. 4.6).

The squall line event (see Fig. 4.5) shows an improvement from the new approach (9.4 with MNN compared to 12.4 km/h average error with U_{rt}). MNN identifies correctly

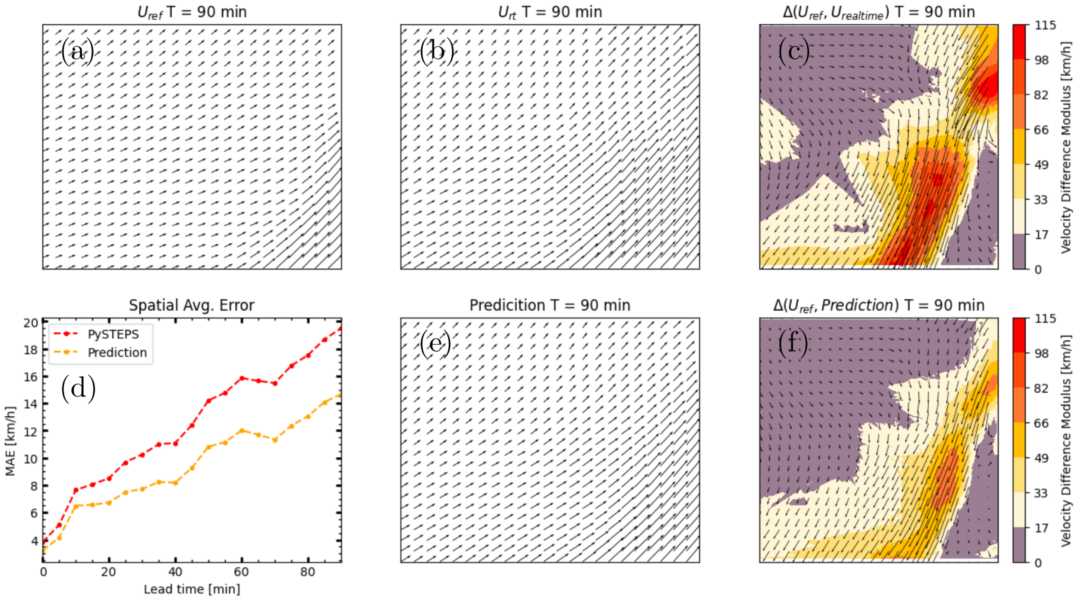


Figure 4.5.: Representation of the event of interest from 2012-06-16-1755 related to squall lines. It contains U_{ref}, U_{rt}, and the prediction at 90 min. (a, b, e), the difference of both with respect to U_{ref} at 90 min. (e, f) and the spatially averaged error over lead time (d).

the areas presenting the biggest issues and modifies them correctly, advecting the field, reducing significantly the error. The band related to the squall line with the most significant problems gets narrowed. Maximum error values decrease by at least 30 km/h which are even greater in the surrounding areas. However, the magnitude of the change is still not sufficient. The temporal evolution of the error looks parallel to PySTEPS but improves even at 0 min and this difference diverges over time.

The isolated convective cell event (see Fig. 4.6) is completely different to the other case. The average error is the same at 3.9 km/h and the temporal evolution almost exactly overlaps with the static approach. The visual inspection of the spatial error shows that the model does not just stick to the given motion field. It seems to identify the areas with issues correctly and tries to make changes although not successfully. The central region in the domain improves but the two other most noticeable regions (southwest and east) have the opposite effect.

MNN shows a dynamical prediction compared to PySTEPS, resulting in motion estimates that are on par with or better than the static PySTEPS estimates. This makes it a worthwhile alternative. In particular, highly structured and stable events substantially benefit. The improvement in some cases can go up to 45% (see C.2.2) and in others, even when the improvement is not homogeneous (see C.1.2), there is an overall benefit. For events with a high impact of growth and decay, where the prediction

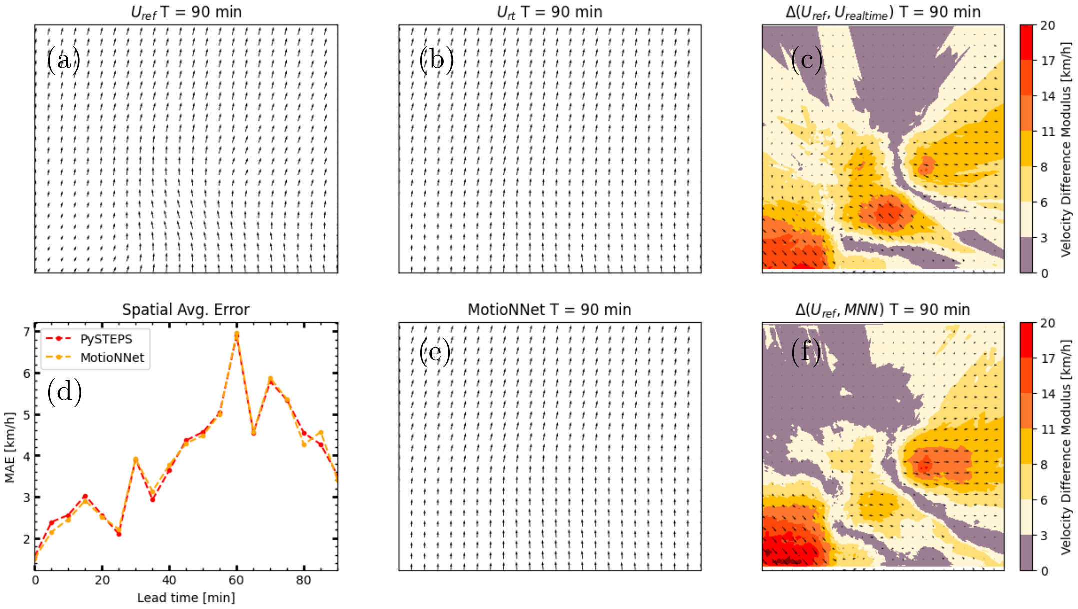


Figure 4.6.: Representation of the event of interest from 2010-07-10-1800 related to isolated convective cells. It contains U_{ref}, U_{rt}, and the prediction at 90 min. (a, b, e), the difference of both with respect to U_{ref} at 90 min. (e, f) and the spatially averaged error over lead time (d).

of the motion is more complex, the performance of MNN is on par with PySTEPS. Hence, MNN provides overall a clear step forward from the original static motion field estimation approach.

4.3 Probabilistic Prediction

4.3.1 Probabilistic Comparison of Motion Estimation

We modified the deterministic models following the methods described in Sec. 2.1 for PySTEPS and Sec. 3.6.1 for the MNN. They are compared using CRPS as described in Section 3.6.2. The differences between MNN and PySTEPS are much smaller in the probabilistic setting than in Section 4.2.1. The MNN models with low dropout probabilities present a smaller spread but higher values in the distribution of the errors than PySTEPS. This is driven by higher CRPS values for the lower quantiles across the ensembles and lower values for the upper quantiles.

The PySTEPS approach (see Fig. 4.7) has the biggest CRPS spread over lead time, but the lowest mean compared to all the other options. For small lead times, MotionNNet and PySTEPS produce very similar error distributions. However, as we move toward

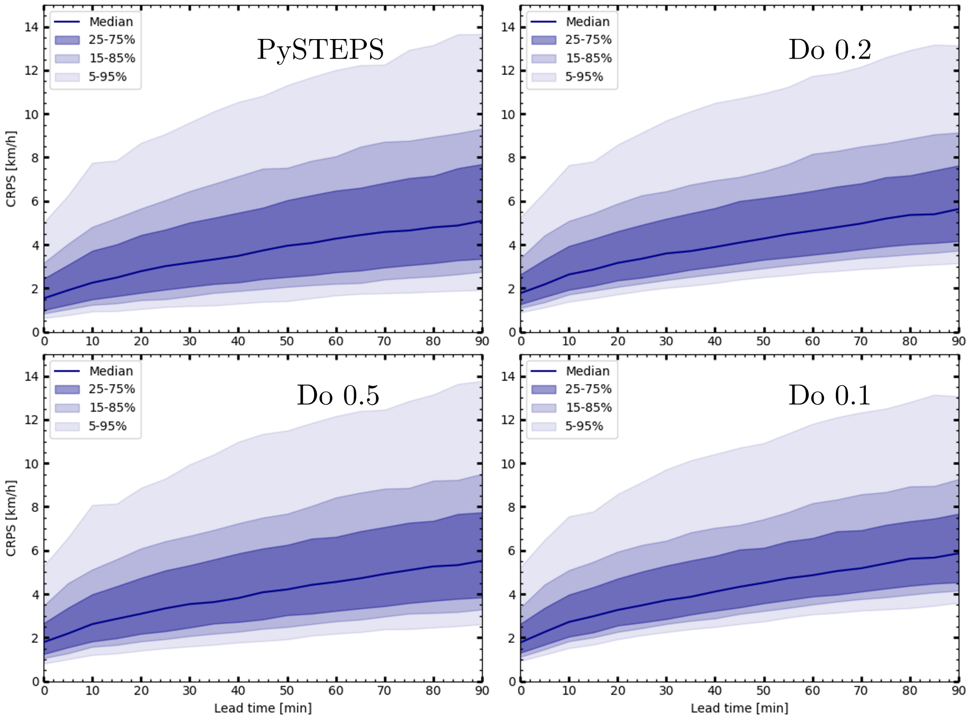


Figure 4.7.: CRPS with lead time comparison on the test set between PySTEPS and the new approach with dropout probabilities of 0.1, 0.2 and 0.5. It includes the average of the x and y components showing different percentiles that indicate the spread of the CRPS.

longer lead times, the differences between the models become more apparent. If we focus on 90 min. (see Fig. 4.7), it is clear that for the lower percentiles, decreasing dropout probabilities lead to higher CRPS values (up to 1.5 km/h higher) compared to PySTEPS, and the opposite occurs in the upper percentiles (up to 0.5 km/h less). This decreases the spread of the error by 20% and displaces the error distribution to larger values on average.

The key to understanding why the errors in PySTEPS behave like this lies in Equation 3.4 (see Section 2.1), where we can see that the perturbations added to the motion field are randomly chosen from a Laplace distribution whose standard deviation increases with lead time. These random perturbations can reach 7-8 km/h for 90 min. lead time (see Sec. 3.6). The problem with this approach is that the same perturbation is applied to the entire spatial domain, independently of the magnitude and local characteristics of the motion field. This does not make a lot of sense from a physical point of view since motion field perturbations are likely to be linked to local changes in rainfall patterns, and locally-varying phenomena such as horizontal convergence, divergence and convection. Realistic ensembles should reflect this idea and only add uncertainty in areas that are susceptible to see significant deviations from Lagrangian persistence.

Therefore, although the average statistics can be better in some cases, the spatial structure of the modifications is relevant and can be problematic, affecting significantly individual ensemble members. It is then important to explore more in-depth specific events.

4.3.2 Ensemble Generation on Events of Interest

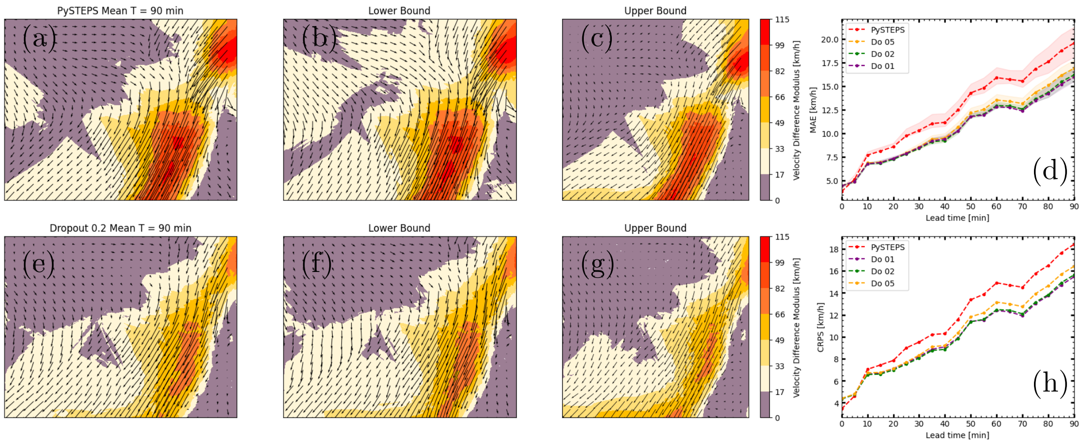


Figure 4.8.: Representation of the event of interest from 2012-06-16-1755 related to squall lines. The top row contains the difference to U_{ref} at 90 min. of the average prediction of the PySTEPS ensemble, the lower and the upper bound (a, b, c). The bottom row is the same for the model with a drop probability of 0.2 (e, f, g). On the right, the spatially averaged MAE of those 3 predictions for PySTEPS and dropouts of 0.1, 0.2 and 0.5 are shown together with the average CRPS for each model (d, h). The shaded area of the MAE plot is the error gap between the bounds.

The events presented in Sec. 3.5.2 are used again to show with greater detail the different error behaviour of each approach. The MNN models, present ensembles where each member is still substantially better than PySTEPS for structured and stable cases (Fig. 4.8). For cases where growth and decay are more prominent, and the velocity errors are smaller, each of the members in PySTEPS performs equally or worse than any of the MNN models (Fig. 4.9, where the red shaded area has a much higher error than the ensemble mean of PySTEPS). They contain unrealistic spatial perturbation patterns, meaning that many individual ensemble members show deviations from the average motion field that are not realistic (i.e., that are very unlikely to occur in reality because the uncertainty of the motion field predictions is likely not to be the same everywhere). The MNN models, even though they perform worse than in the deterministic case, have realistic ensembles where the problematic areas of the prediction are clearly identified.

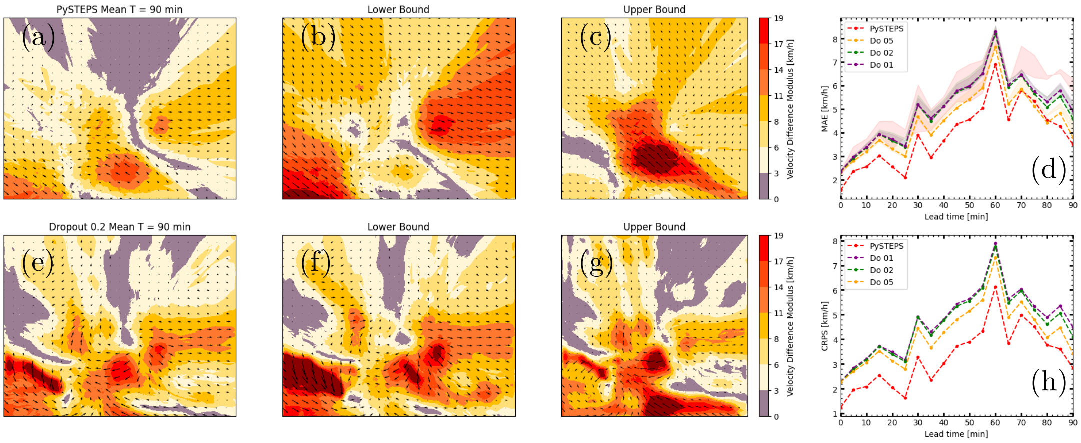


Figure 4.9.: Representation of the event of interest from 2010-07-10-1800 related to isolated convective events. The top row contains the difference to U_{ref} at 90 min. of the average prediction of the PySTEPS ensemble, the lower and the upper bound (a, b, c). The bottom row is the same for the model with a drop probability of 0.2 (e, f, g). On the right, the spatially averaged MAE of those 3 predictions for PySTEPS and dropouts of 0.1, 0.2 and 0.5 are shown together with the average CRPS for each model (d, h). The shaded area of the MAE plot is the error gap between the bounds.

The squall line event (see Fig. 4.8) is very similar to the deterministic analysis. The temporal evolution of the error with dropout is slightly worsened. The error spread increases for both the dropout and PySTEPS models over time with a bigger spread for PySTEPS, common to other structured and stable cases (see Ap.C). The mean lies in the middle of the spread, therefore not improving compared to individual members. The CRPS behaviour is almost identical to the MAE. Spatially, the mean does not improve the deterministic prediction but is quite similar, with a larger error region to the left of the domain. The bounds are close to the mean and for the areas with larger errors, the deterministic prediction lies between them. For the areas with low errors differences can be seen but are much smaller than in the PySTEPS bounds. The spatial structure of the field is well preserved in both PySTEPS and MNN without unrealistic changes.

For the isolated convective cell event (see Fig. 4.9), the average prediction of PySTEPS remains the same as in the deterministic case, but the dropout models perform worse by approximately 1 km/h. The spread of the MNN dropout models is small, but the mean is more robust than individual members, albeit worse than the deterministic prediction. Normally with ensembles, the ensemble mean is supposed to be better than the deterministic run, indicating that none of the options are yet refined or optimal enough. PySTEPS has a larger spread than the dropout cases, and its members generally perform equal to or worse than the dropout methods. The CRPS indicates better performance for PySTEPS but misses information related to the spatial distribution of the errors. The ensemble members of PySTEPS can be very far apart (with members showing NW motion errors and other members indicating SE errors), which you can see in the lower bound and upper bound of Fig. 4.9. This indicates complementary errors due to the random sampling in the PySTEPS ensemble generation procedure. The dropout ensemble does not perform as well as the deterministic case on average. The most challenging regions are identified, and the model attempts to make changes. These changes result in large error blobs in regions with significant growth and dissipation processes, yet the spatial structure is more consistent in the members (which can be positive but also reduces the spread), which is the main difference from PySTEPS.

The CRPS description presented a higher error distribution for MNN compared to PySTEPS. Nevertheless, when exploring specific events, PySTEPS ensemble generation can produce unrealistic members. This is due to perturbations that are not based on the underlying context, but that are independent of the magnitude and local characteristics of the motion field. MNN models produce better predictions in structured and stable events. They also identify problematic areas accurately and generate more realistic ensemble members overall. In events with large motion errors, the spread in both methods is too small but affects MNN more intensely. This shows that uncertainty quantification in convective events is still a big challenge, where the magnitude (especially for MNN) and the spatial distribution of the perturbations (focusing on PySTEPS) in the ensembles play a very important role.

5 Conclusions

5.1 Answers to Research Questions

The work can be summarized in three main parts: in the first part, an in-depth analysis and evaluation of motion field estimates and forecasts in PySTEPS was conducted. In the second part, a new machine-learning model, MotioNNet, for predicting motion fields based on a modified implementation of the NowcastNet architecture was trained. In the third part, the ensemble predictions of the two models were analyzed and compared to each other.

The main findings are as follows:

1. **What are the current errors/uncertainties associated with estimating motion fields in PySTEPS?** The results show errors already existing at the beginning of the prediction, approximately 1-3 km/h on average, that increase and spread with lead time, 4-8 km/h at 90 min. on average. This can lead to average displacement errors of 3-8 km, which means that the localized misplacements of rain cells can be much higher, up to tens of km. Using two events of interest, we show that the advection of the motion field for highly structured and stable events is crucial for an improved prediction. Isolated convective cells impose larger problems due to the connection between the growth and decay of precipitation and motion field dynamics.
2. **How can the PySTEPS motion field predictions be improved using machine learning?** We present a new Deep Learning model, MotioNNet, that predicts changes in the PySTEPS motion field. The model was trained on 7,500 events, validated on 1,250 events and tested on 1,250 events that contained high-intensity rainfall and motion error in the Netherlands. The inputs used are rainfall fields and PySTEPS motion field estimation. MotioNNet improves the existing motion field estimation approach by approximately 13% on average, affecting the most problematic events. On specific events, this improvement can go up to 45 %. The model identifies regions that are an issue for PySTEPS and in general, improves the motion estimate there. The model predicts the advection of the motion but in general not as intense in magnitude as in the reference. We

expect that our new method using dynamic error predictions of motion fields can strengthen some of the weaknesses of extrapolation-based methods where the steady-state assumption does not hold.

3. **How can machine learning models be used to generate ensembles of motion field predictions, and how do these ensembles compare to the ones currently implemented in PySTEPS?** SpatialDropout and Monte Carlo Dropout are included in MotioNNet to create probabilistic predictions. This combination makes better predictions in structured and stable events. It accurately identifies problematic areas and generates realistic ensemble members. PySTEPS ensemble generation is theoretically limited, providing a better spread for events with large errors but unrealistic members where low errors are present. The new method introduces new features, like accurately identifying the spatial distribution of uncertainty, lacking in the current PySTEPS approach. Unfortunately, it does not yet provide enough ensemble spread, resulting in an insufficient captured forecast uncertainty.

5.2 Recommendations

These points suggest new ways forward. One option would be improving the model by modifying the existing architecture or including new options like ConvLSTMs, Diffusion models or Transformers, which could provide better predictions. New and different ways of introducing dropout in MNN can still be explored, like restricting dropout to the deeper feature maps while increasing the dropout rate. Some of these alternative architectures also allow modifications to become probabilistic, especially Diffusion models, producing better deterministic predictions and generating better ensembles.

These approaches can require large efforts and complexities (e.g. difficulties in the implementation, training costs, data requirements, etc.). Therefore, another alternative could be modifying the current PySTEPS ensemble generation procedure. Developing new ways of introducing context-dependent perturbations, might be interesting to generate ensembles that adapt to each event. These can be based on, for example, magnitude, motion and rain spatial structure or uncertainty in the prediction. For that, a combination of Deep Learning algorithms could help in the spatial identification of uncertainty (like U-Nets) but also in the required magnitude of the perturbations (models with spatial and temporal inductive biases could be interesting to explore).

Furthermore, the reference used in this project, key in all the analyses performed, is limited. There are no observations of motion but only wind observations. It would be crucial to find a systematic way of assessing the quality of a motion method. Nonetheless, the current evaluation and motion field dataset for the 10,000 events can provide a good benchmark dataset to train future motion field estimation models. The events of interest can be further analyzed to identify more case studies which could, for example, be part of a benchmark not only of motion but also rain. Outside the dataset, the removed events can also give more clues on the possible issues that observations can present, potentially using motion fields for detecting clutter or outliers.

This study demonstrated that integrating deep learning with traditional methods can improve the accuracy of problematic areas in the rainfall nowcasting chain. By incorporating the MotioNNet model into PySTEPS, we achieved more precise motion field forecasts, particularly for complex weather patterns. This hybrid approach leverages the interpretability and reliability of traditional models alongside the adaptability and high performance of machine learning. The success of this integration suggests that future projects should continue to develop and refine these combined methodologies. Advancements in this area promise to enhance the accuracy and reliability of rainfall nowcasting, which is crucial for society, especially in the context of increasingly extreme high-intensity events.

Bibliography

[1] Y. Zhang, M. Long, K. Chen, L. Xing, R. Jin, M. Jordan, and J. Wang, “Skilful nowcasting of extreme precipitation with nowcastnet,” *Nature*, vol. 619, pp. 1–7, 07 2023.

[2] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using convolutional networks,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 648–656, 2015.

[3] Y. Wang, E. Coning, A. Harou, W. Jacobs, P. Joe, L. Nikitina, R. Roberts, J. Wang, J. Wilson, A. Atencia, B. Bica, B. Brown, S. Goodmann, A. Kann, P. W. Li, I. Monterio, F. Schmid, A. Seed, and J. Sun, *Guidelines for Nowcasting Techniques*. 11 2017.

[4] R. Imhoff, *Rainfall nowcasting for flood early warning*. internal phd, wu, Wageningen University, Netherlands, 2022. WU thesis 8244 Includes bibliographical references. - With summaries in English and Dutch.

[5] A. Smith, “The atlas of mortality and economic losses from weather, climate and water extremes (1970–2019),” 09 2021.

[6] B. van den Hurk, P. Siegmund, A. K. Tank, J. Attema, A. Bakker, J. Beersma, J. Bessembinder, R. Boers, T. Brandsma, H. van den Brink, S. Drijfhout, H. Eskes, R. Haarsma, W. Hazeleger, R. Jilderda, C. Katsman, G. Lenderink, J. Loriaux, E. van Meijgaard, T. van Noije, G. van Oldenborgh, F. Selten, P. Siebesma, A. Sterl, H. de Vries, M. van Weele, R. de Winter, and G.-J. van Zadelhoff, “Knmi’14: Climate change scenarios for the 21st century – a netherlands perspective,” p. 120, 01 2014.

[7] A. Seed, “A dynamic and spatial scaling approach to advection forecasting,” *Journal of Applied Meteorology - J APPL METEOROL*, vol. 42, pp. 381–388, 03 2003.

[8] N. Bowler, C. Pierce, and A. Seed, “Steps: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled nwp,” *Quarterly Journal of the Royal Meteorological Society*, vol. 132, pp. 2127 – 2155, 01 2007.

[9] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge, R. Prudden, A. Mandhane, A. Clark, A. Brock, K. Simonyan, R. Hadsell, N. Robinson, E. Clancy, A. Arribas, and S. Mohamed, "Skilful precipitation nowcasting using deep generative models of radar," *Nature*, vol. 597, p. 672–677, Sept. 2021.

[10] S. Pulkkinen, D. Nerini, A. A. Pérez Hortal, C. Velasco-Forero, A. Seed, U. Germann, and L. Foresti, "Pysteps: an open-source python library for probabilistic precipitation nowcasting (v1.0)," *Geoscientific Model Development*, vol. 12, no. 10, pp. 4185–4219, 2019.

[11] C. Short and J. Petch, "Reducing the spin-up of a regional nwp system without data assimilation," *Quarterly Journal of the Royal Meteorological Society*, vol. 148, 05 2022.

[12] J. Sun, M. Xue, J. W. Wilson, I. Zawadzki, S. P. Ballard, J. Onvlee-Hooimeyer, P. Joe, D. M. Barker, P.-W. Li, B. Golding, M. Xu, and J. Pinto, "Use of nwp for nowcasting convective precipitation: Recent progress and challenges," *Bulletin of the American Meteorological Society*, vol. 95, no. 3, pp. 409 – 426, 2014.

[13] R. Imhoff, C. Brauer, K.-J. Heeringen, R. Uijlenhoet, and A. Weerts, "Large-sample evaluation of radar rainfall nowcasting for flood early warning," *Water Resources Research*, vol. 58, 03 2022.

[14] R. Prudden, S. Adams, D. Kangin, N. Robinson, S. Ravuri, S. Mohamed, and A. Arribas, "A review of radar-based nowcasting of precipitation and applicable machine learning techniques," 2020.

[15] S. Ryu, G. Lyu, Y. Do, and G. Lee, "Improved rainfall nowcasting using burgers' equation," *Journal of Hydrology*, vol. 581, p. 124140, 09 2019.

[16] U. Germann, I. Zawadzki, and B. Turner, "Predictability of precipitation from continental radar images. part iv: Limits to prediction," *Journal of the Atmospheric Sciences*, vol. 63, no. 8, pp. 2092 – 2108, 2006.

[17] T. Palmer and J. Räisänen, "Quantifying the risk of extreme seasonal precipitation events in a changing climate," *Nature*, vol. 415, no. 6871, pp. 512–514, 2002.

[18] D. S. Richardson, "Skill and relative economic value of the ecmwf ensemble prediction system," *Quarterly Journal of the Royal Meteorological Society*, vol. 126, no. 563, pp. 649–667, 2000.

[19] L. Foresti, M. Reyniers, A. Seed, and L. Delobbe, "Development and verification of a real-time stochastic precipitation nowcasting system for urban hydrology in belgium," *Hydrology and Earth System Sciences*, vol. 20, pp. 505–527, 01 2016.

[20] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W. K. Wong, and W.-c. WOO, "Deep learning for precipitation nowcasting: A benchmark and a new model," 06 2017.

[21] S. Agrawal, L. Barrington, C. Bromberg, J. Burge, C. Gazen, and J. Hickey, "Machine learning for precipitation nowcasting from radar images," *CoRR*, vol. abs/1912.12132, 2019.

[22] C. Kaparakis and S. Mehrkanoon, "Wf-unet: Weather data fusion using 3d-unet for precipitation nowcasting," *Procedia Computer Science*, vol. 222, pp. 223–232, 2023. International Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023).

[23] G. Ayzel, T. Scheffer, and M. Heistermann, "Rainnet v1.0: a convolutional neural network for radar-based precipitation nowcasting," *Geoscientific Model Development*, vol. 13, no. 6, pp. 2631–2644, 2020.

[24] K. Trebing, T. Staczyk, and S. Mehrkanoon, "Smaat-unet: Precipitation nowcasting using a small attention-unet architecture," *Pattern Recognition Letters*, vol. 145, pp. 178–186, 2021.

[25] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," 1999.

[26] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *International Joint Conference on Artificial Intelligence*, 1981.

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] E. Ruzanski, V. Chandrasekar, and Y. Wang, "The casa nowcasting system," *Journal of Atmospheric and Oceanic Technology - J ATMOS OCEAN TECHNOL*, vol. 28, pp. 640–655, 05 2011.

[29] S. Laroche and I. Zawadzki, "A variational analysis method for retrieval of three-dimensional wind field from single-doppler radar data," *Journal of Atmospheric Sciences*, vol. 51, pp. 2664–2682, 08 1994.

[30] U. Germann and I. Zawadzki, "Scale-dependence of the predictability of precipitation from continental radar images. part i: Description of the methodology," *Monthly Weather Review*, vol. 130, no. 12, pp. 2859 – 2873, 2002.

[31] M. Keller and W. Coventry, "Quantifying and addressing parameter indeterminacy in the classical twin design," *Twin research and human genetics : the official journal of the International Society for Twin Studies*, vol. 8, pp. 201–13, 07 2005.

[32] M. L. Johnson, "Parameter correlations while curve fitting," in *Numerical Computer Methods, Part C*, vol. 321 of *Methods in Enzymology*, pp. 424–446, Academic Press, 2000.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[35] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *Proceedings of The 33rd International Conference on Machine Learning*, 06 2015.

[36] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: A regularization method for convolutional networks," 2018.

[37] T. A. Brown, "Admissible scoring systems for continuous distributions.," 1974.

[38] J. E. Matheson and R. L. Winkler, "Scoring rules for continuous probability distributions," *Management Science*, vol. 22, pp. 1087–1096, 1976.

[39] H. Hersbach, "Decomposition of the continuous ranked probability score for ensemble prediction systems," *Weather and Forecasting - WEATHER FORECAST*, vol. 15, pp. 559–570, 10 2000.

[40] Sven van Os, "Precipitation Nowcasting using a Generative Adversarial Network." `http://resolver.tudelft.nl/uuid:35a11963-4524-442f-97a1-016694b3813f`, Master's Thesis, 2024, TU Delft Repository.

[41] J. S. Marshall and W. M. K. Palmer, "The distribution of raindrops with size," *Journal of Atmospheric Sciences*, vol. 5, no. 4, pp. 165 – 166, 1948.

[42] R. Zhong, S. Xu, F. Huang, and X. Wu, "Reasons for the weakening of tropical depressions in the south china sea," *Monthly Weather Review*, vol. 148, no. 8, pp. 3453 – 3469, 2020.

[43] Delft High Performance Computing Centre (DHPC), "DelftBlue Supercomputer (Phase 2)." `https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2`, 2024.

[44] R. Rotunno, J. Klemp, and M. Weisman, "A theory for strong, long-lived squall lines," *Journal of The Atmospheric Sciences - J ATMOS SCI*, vol. 45, pp. 463–485, 02 1988.

ChatGPT and Bard have been used in this master thesis for coding purposes and to rewrite and improve vocabulary and grammar.

A Events

A.1 Relevant events

’2020-02-09-1550’,’2022-12-30-1225’,’2012-06-16-1755’,’2012-01-03-1410’,’2010-07-10-1800’,’2010-07-14-1530’,’2009-05-26-0315’,’2009-05-26-0010’,’2020-02-16-1715’,’2021-10-21-0125’,’2015-01-09-1610’,’2017-06-09-0720’,’2012-06-16-1020’,’2008-03-01-0145’,’2008-06-08-1355’,’2008-06-08-1700’,’2008-07-28-0145’,’2020-02-20-1705’,’2019-06-04-1950’,’2013-11-04-0225’,’2012-05-23-1530’

A.2 Removed events with cap

’2020-08-30-1535’,’2021-07-18-0005’,’2012-05-27-1530’,’2019-08-31-1450’,’2017-05-23-0125’,’2019-09-25-1730’,’2021-07-11-1120’,’2022-07-22-2055’,’2018-01-17-1905’,’2021-07-07-0855’,’2021-07-08-1025’,’2020-12-24-0935’,’2016-09-01-0635’,’2019-07-24-0635’,’2010-09-20-1855’,’2010-03-29-0200’,’2021-06-30-1305’,’2017-11-02-1430’,’2019-09-27-1150’,’2019-09-22-1600’,’2017-07-11-0115’,’2022-01-03-1155’,’2019-02-04-1105’,’2020-08-06-2230’,’2021-11-07-1245’,’2017-10-03-1235’,’2020-08-21-1220’,’2017-03-11-0710’,’2020-09-06-1900’,’2014-09-28-0140’,’2021-08-27-1330’,’2021-06-26-1205’,’2021-08-22-2045’,’2021-06-28-0900’,’2020-11-15-2040’,’2015-08-19-1250’,’2021-02-05-1535’,’2020-02-16-2025’,’2016-01-28-0100’,’2017-07-07-2225’,’2016-09-07-2125’,’2015-01-09-1915’,’2012-04-27-0125’,’2018-01-27-1715’,’2010-12-22-2340’,’2021-10-15-0535’,’2020-09-21-0455’,’2020-06-23-0400’,’2021-08-20-0550’,’2014-09-03-0410’,’2016-05-11-1350’,’2009-02-09-2125’,’2020-10-10-0910’,’2016-07-20-0525’,’2020-06-17-0355’,’2022-08-05-0355’,’2020-10-05-1140’,’2017-03-19-1715’,’2020-11-15-1005’,’2022-07-08-2335’,’2008-08-03-0010’,’2017-06-02-0240’,’2017-10-03-0615’,’2012-06-30-0710’,’2021-08-12-1700’,’2021-08-27-0920’,’2019-09-27-0845’,’2017-06-07-1340’,’2017-12-10-0815’,’2020-10-21-1905’,’2016-09-22-0245’,’2021-09-11-1055’,’2021-08-26-1710’,’2021-11-07-1550’,’2016-09-14-0135’,’2017-10-19-2030’,’2019-08-11-1605’,’2022-09-23-0255’,’2017-05-27-2230’,’2011-04-24-1830’,’2022-01-08-1215’,’2021-10-22-1125’,’2009-11-01-2130’,’2020-08-24-2045’,’2009-11-01-1455’,’2020-07-26-1820’,’2020-09-05-0220’,’2021-10-22-1430’,’2011-02-04-0245’,’2012-01-04-1945’,’2021-07-14-0705’,’2020-08-25-1830’,’2016-02-09-0815’,’2020-02-23-0915’,’2020-08-15-1840’,’2017-07-08-0155’,’2020-09-06-0505’,’2020-09-07-0955’,’2017-02-27-0510’,’2019-06-21-1040’,’2020-08-22-1500’,’2021-11-01-1440’,’2009-11-01-1150’,’2020-08-20-2010’,’2021-09-17-1135’,’2019-

07-26-1540’,’2021-09-11-1400’,’2020-08-18-1540’,’2020-07-26-0925’,’2020-08-03-0850’,’2021-07-05-1215’,’2022-06-07-1805’,’2010-11-04-0110’,’2016-07-19-0510’,’2021-07-26-0805’,’2022-01-05-1010’,’2012-08-18-0335’,’2017-10-20-2025’,’2021-03-31-0820’,’2017-10-27-0240’,’2009-05-28-0900’,’2021-07-18-0310’,’2021-07-13-0920’,’2017-07-03-0335’,’2017-05-29-0125’,’2017-10-03-0255’,’2019-09-10-0310’,’2020-04-27-0330’,’2013-08-28-2055’,’2017-06-04-0945’,’2018-01-17-1320’,’2017-11-16-1230’,’2020-11-15-1310’,’2017-03-03-1635’,’2021-04-19-0810’,’2013-11-06-1500’,’2017-12-07-1815’,’2022-08-25-1605’,’2021-02-25-1810’,’2010-12-19-1750’,’2020-10-23-1820’,’2009-08-21-1150’,’2021-08-20-1855’,’2020-09-15-0850’,’2009-11-02-0035’,’2019-09-06-0135’,’2021-06-04-1250’,’2021-09-14-1325’,’2017-06-11-0410’,’2017-07-31-1110’,’2011-03-31-0740’,’2018-01-10-1410’,’2021-04-12-0840’,’2020-09-25-1520’,’2020-09-07-0555’,’2022-04-07-0650’,’2020-08-30-2125’,’2008-11-18-1355’,’2013-04-15-0830’,’2014-09-28-0650’,’2017-09-19-0950’,’2017-08-31-0925’,’2022-08-18-1350’,’2021-10-06-1055’,’2017-05-21-2330’,’2017-05-22-0235’,’2009-11-01-1800’,’2021-08-20-1550’

A.3 Examples: radar artefacts and outliers

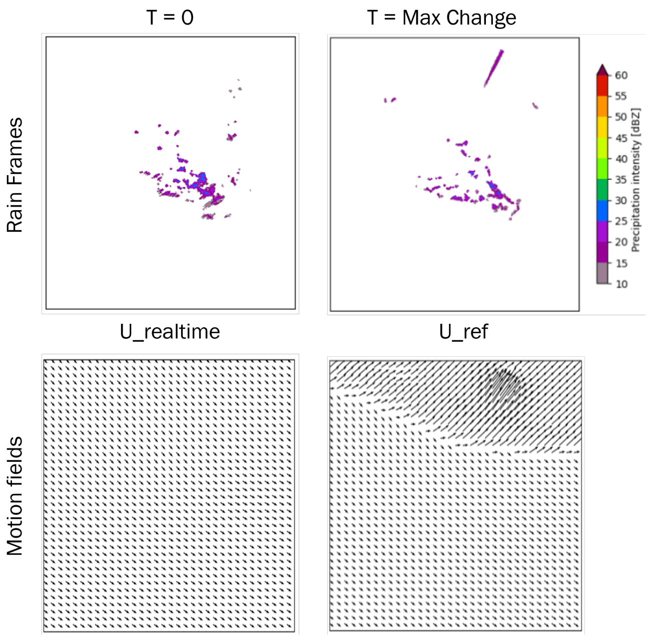


Figure A.1.: Example event with radar artefacts.

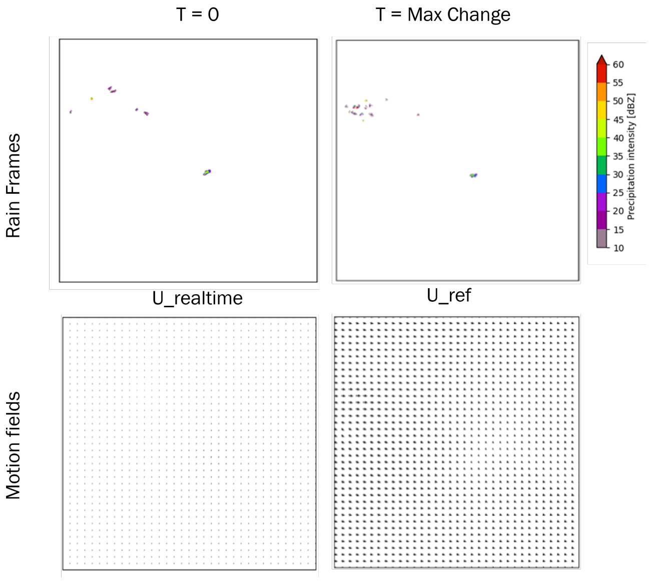


Figure A.2.: Example event with clutter.

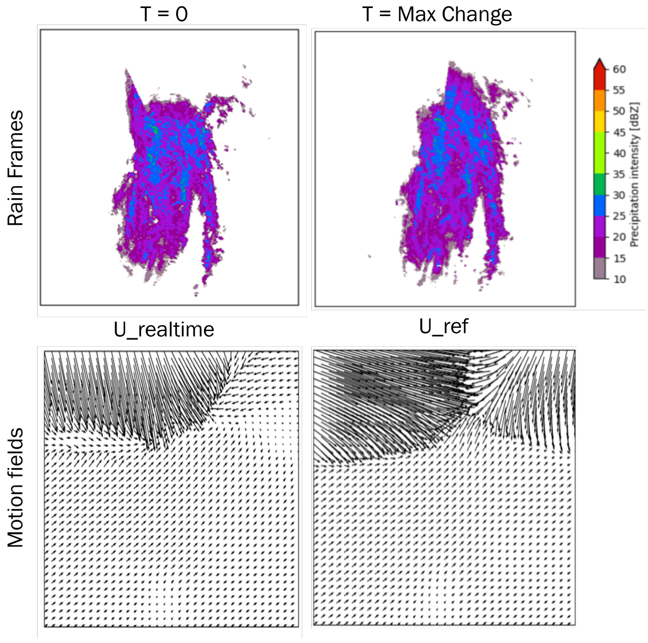


Figure A.3.: Example event removed with the cap.

B Velocity and Displacement Errors

B.1 Velocity Errors with all “valid” events

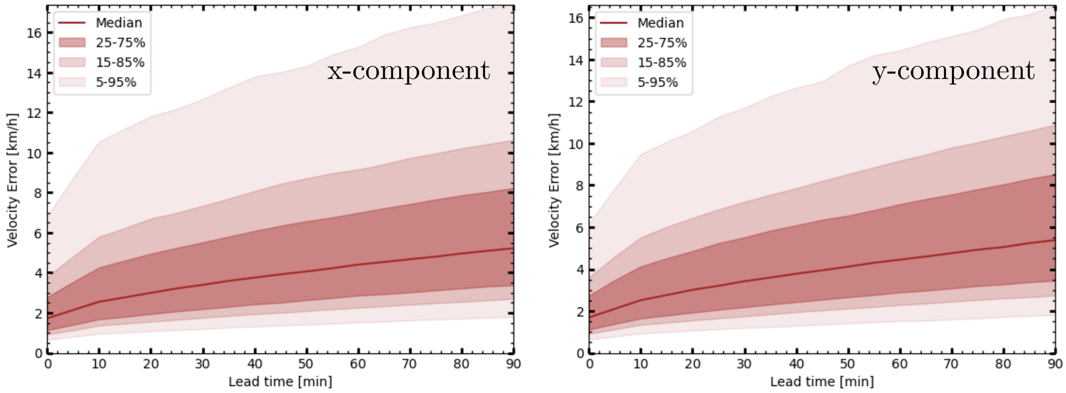


Figure B.1.: MAE of U_{rt} vs U_{ref} for all the cases that contain rain including both the x (left) and y (right) components. Different percentiles are shown indicating the spread of the error in the different events.

Figures 4.1 and B.1 show similar distributions. However, filtering of Sec. 3.4 results in a lower spread and a higher median. This indicates that extreme motion field errors, caused by erroneous rainfall fields, are filtered out and relevant cases with high motion errors are selected. This is expected as the cap aims to remove outliers and equation 3.4 sets the focus on motion error, shifting the distribution slightly to cases with larger differences.

B.2 Displacement Errors

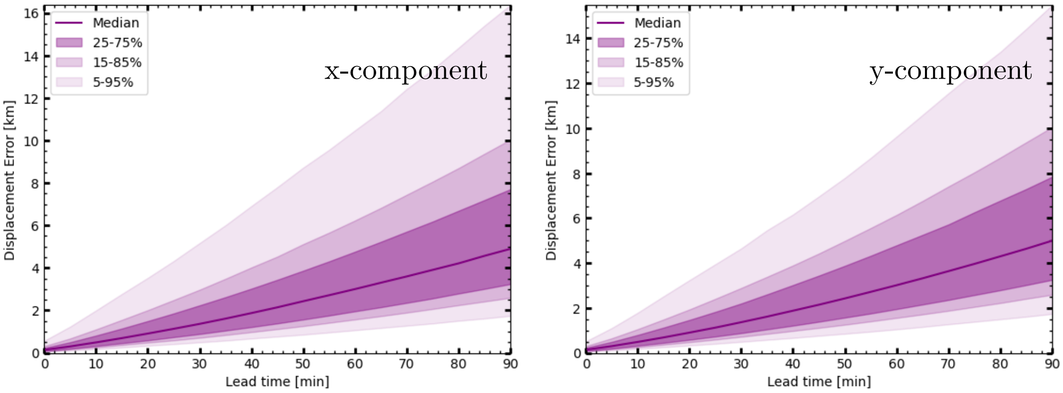


Figure B.2.: MAE of the displacement error associated to $U_{realtime}$ vs U_{ref} for the top 10000 events selected based on eq. 3.4 with a at 35 km/h including both the x (left) and y (right) components. Different percentiles are shown indicating the spread of the error in the different events.

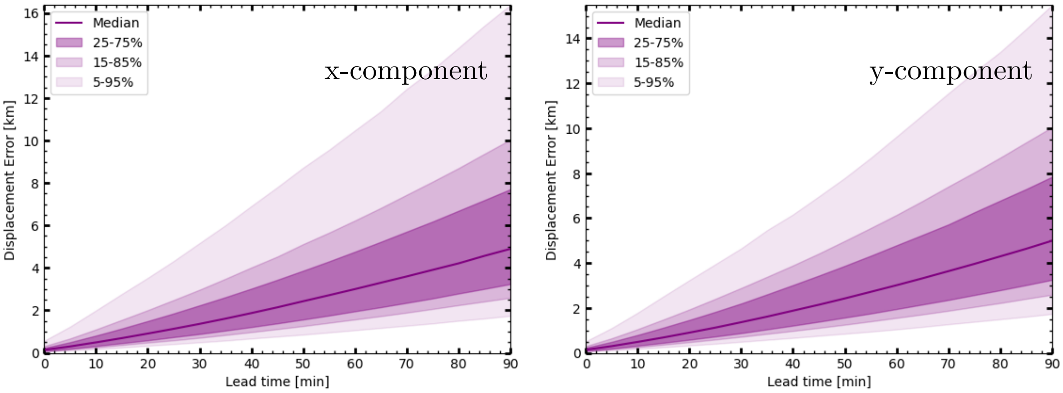


Figure B.3.: MAE of the displacement error associated to $U_{realtime}$ vs U_{ref} for all the cases that contain rain including both the x (left) and y (right) components. Different percentiles are show indicating the spread of the error in the different events.

C Events of Interest Results

C.1 Extra Squall Line Event

This event from 2012-01-03-1410 presents a squall line embedded inside a larger rain field, all of it moving east. The squall line moves with the rain field to the east but also slightly southward. Over 90 min, the rain before the line dissipates and the one at the back increases in intensity and slightly move north. This embedding of the squall line inside the larger rain fields prevents U_{ref} from identifying its correct movement, as the surrounding rain becomes predominant in the motion calculation.

C.1.1 PySTEPS Analysis on Extra Squall Line

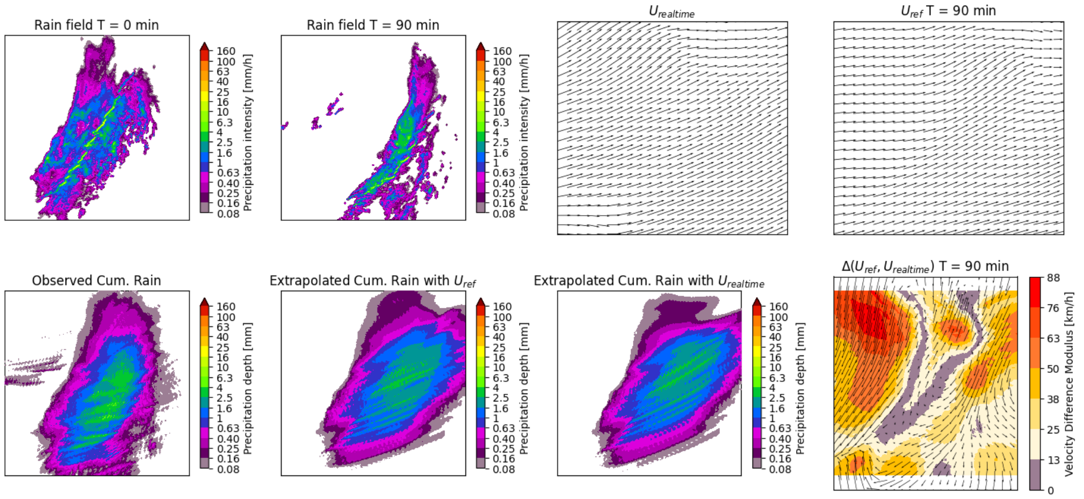


Figure C.1.: Representation of the event of interest from 2012-01-03-1410 related to squall lines. It contains the rain observation at 0 and 90 min., U_{rt}, U_{ref} at 90 min., their difference at 90 min. and the observed and extrapolated cumulative rainfall over 90 min. using the static and the dynamic approach for the latter.

Eventhough U_{ref} optical flow calculation do not fully capture the correct displacement of the squall line, in the extrapolation based on U_{ref}, the structure of the segments related to the squall line are more accurate than the nowcast using U_{rt}, showing more horizontal and centred stripes (see Fig. C.1). The biggest difference between the motion fields comes from an area with no rain, as this area changes significantly once the rain moves east but there is a clear segment of error identified where the rain is located at the end of the event.

C.1.2 Motion Field Estimation on Squall Line Event

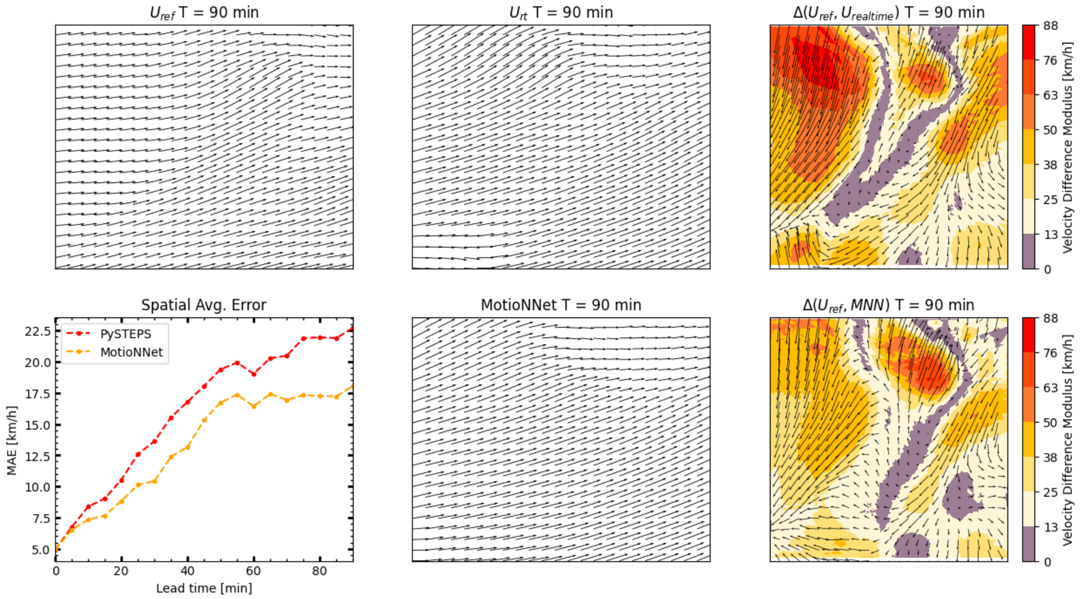


Figure C.2.: Analysis of the event of interest from 2012-01-03-1410 related to squall lines. It contains U_{ref}, U_{rt}, and the prediction at 90 min., the difference of both with respect to U_{ref} at 90 min. and the spatially averaged error over lead time.

In the extra squall line event (see Fig. C.2), MNN overall provides an improvement in the prediction (average error 13.2 km/h) compared to U_{rt} (average error 15.98 km/h). This can be seen in the temporal evolution which starts being equal and diverges with lead times but is not homogeneous in space. Focusing on 90 min. lead time, it can be seen that, in general, the regions with bigger errors are identified and improved. The magnitude of this can go up to several tens of km/h. The only exception is the north middle section that gets worsened, as the models seem to overextend the horizontal section at the NE westward. The regions without rain in the west and the areas covered by rain at the end of the prediction are the ones with the biggest improvement, and in the centre, a small advection for motion field can be appreciated to the east.

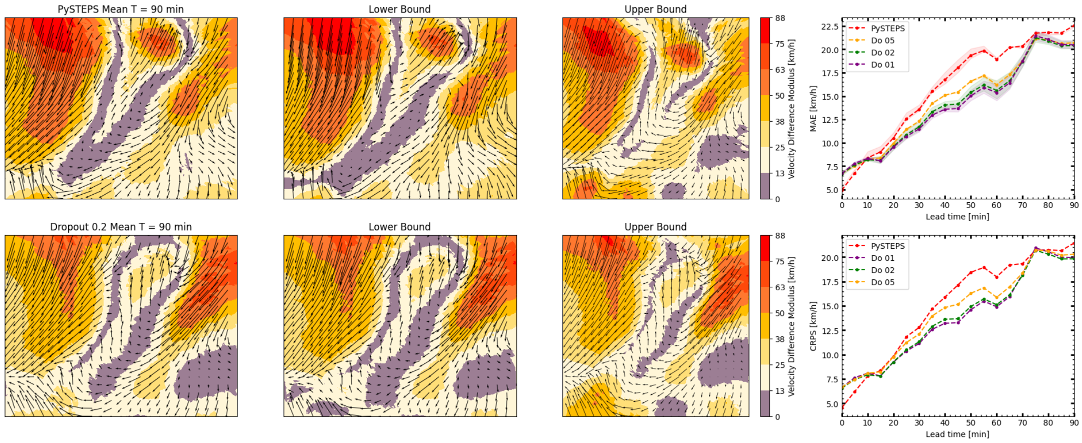


Figure C.3.: Representation of the event of interest from 2012-01-03-1410 related to squall lines. The top row contains the difference to U_{ref} at 90 min. of the average prediction of the PySTEPS ensemble, the lower and the upper bound. The bottom row is the same for the model with a drop probability of 0.2. On the right, the spatially averaged MAE of those 3 predictions for PySTEPS and dropouts of 0.1, 0.2 and 0.5 are shown together with the average CRPS for each model. The shaded area of the MAE plot is the error gap between the bounds.

C.1.3 Probabilistic Comparison on Squall Line Event

For the probabilistic results on the extra squall line event (see Fig. C.3), there are changes compared to the deterministic results. The temporal evolution of the error gets worsened for all the dropout probabilities. In all the events, the model with 0.5 dropout lies in between PySTEPS and the other cases which are very close together. For the first 10 minutes, PySTEPS performs better, as the new ensemble modifies too much the starting PySTEPS prediction (not shown but visually inspected). From 10 to 65 min. the performance is very similar but around 75 min there is a peak that makes PySTEPS and the dropout prediction equal. This is due to the sudden changes in the reference created by noisy fluctuations at the top of the domain in the rainfall field (not shown but visually inspected). The deterministic prediction gets less affected but the ensemble predictions have bigger issues, penalizing the error. The rest of the domain presents clear improvements with a much closer average error to the deterministic prediction. The spread of the averaged error increases in the dropout cases with time, whereas for PySTEPS, it increases and decreases. This is produced by changes in space that compensate for each other, making the real spread per pixel much bigger. The mean lies approximately in the middle of the spread for both approaches, which implies that the mean is not a better prediction. On the spatial side, one of the main changes is that the model correctly identifies the areas with bigger errors and makes modifications but now the centre blob is substantially better, the large area at the left is improved less and the section at the right of the image covered by rain at the end of the prediction is worsened. The bounds with respect to the mean are quite similar for both PySTEPS and dropout, where the spatial structure is very similar in both and well preserved, although the dropout case is more consistent, especially looking at regions with lower error but making smaller modifications as well.

C.2 Cyclonic Rotation Event

Cyclonic storms are highly structured and stable systems. They have a characteristic cyclonic rotation combined with a displacement to the east. This movement is very structured but the combined dynamic in its movement makes it hard to capture in non-dynamic approaches.

C.2.1 PySTEPS Analysis on Cyclonic Event

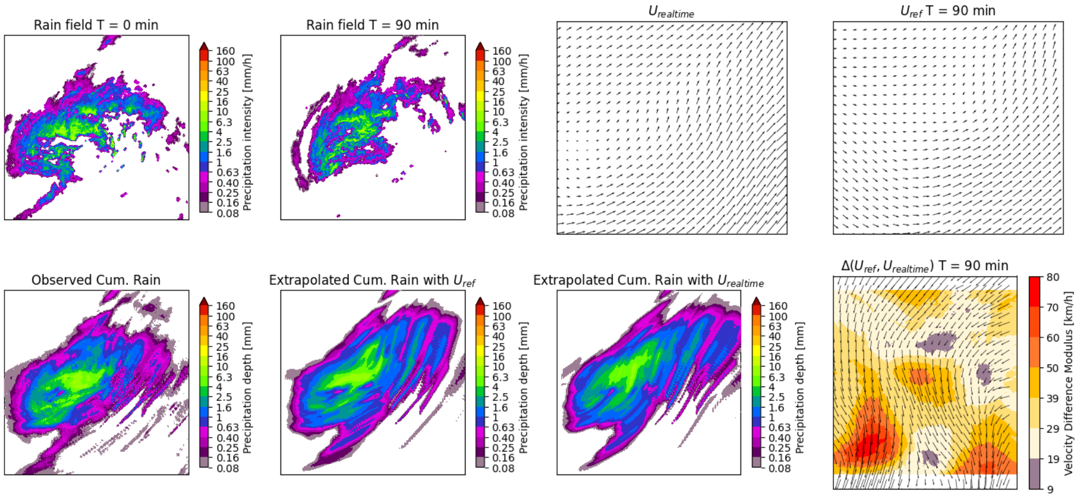


Figure C.4.: Representation of the event of interest from 2021-10-21-0125 related to cyclonic rotations. It contains the rain observation at 0 and 90 min., U_{rt}, U_{ref} at 90 min., their difference at 90 min. and the observed and extrapolated cumulative rainfall over 90 min. using the static and the dynamic approach for the latter.

As the storm moves (see Fig. C.4), the southwards movement at the west of the storm, the slower centre and the northward movement at the east have to displace eastward as a whole. U_{rt} fails to capture this and only portrays the rotation in the past at a single location with no translation of the rotation. In the extrapolation done with U_{ref} compared to U_{rt} (see Fig. C.4) the left side of the storm deepens south more than the static case and the horizontal distribution of the interior cell with the highest intensity is well preserved, very similarly to the observation. The regions at the NE seem more erratic but this is due to big changes of the growth and decay that are not considered in these nowcasts. It is clear that in this example, similarly to the squall line cases, the magnitude of the error in the motion field is significant.

C.2.2 Motion Field Estimation on Cyclonic Event

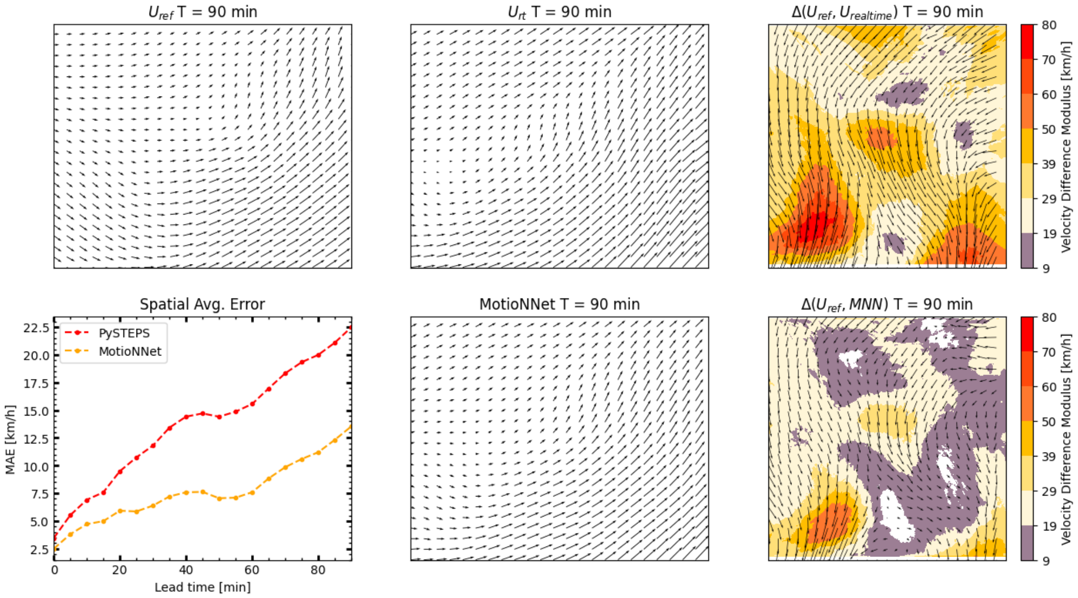


Figure C.5.: Representation of the event of interest from 2021-10-21-0125 related to cyclonic rotations. It contains U_{ref}, U_{rt}, and the prediction at 90 min., the difference of both with respect to U_{ref} at 90 min. and the spatially averaged error over lead time.

The cyclonic rotation event (see Fig. C.5) shows the biggest improvement out of all (7.6 compared to 13.7 km/h average error). The temporal evolution is the same as in the previous examples but the divergence over time is greater. Spatially, the improvement can be seen everywhere. There is an advection of the entire movement to the east although not as prominent as in the reference. The same happens with the characteristic structure of the rotation, which is preserved and captured in the model although the southward movement at the west and the static centre have still issues.

C.2.3 Probabilistic Comparison on Cyclonic Event

The rotation event (see Fig. C.6) results show the same behaviour as the squall line event in the main results. The temporal evolution is worse in the dropout models than the deterministic prediction, but still better than any of the PySTEPS members. Although wider, the spread has a similar behaviour and the same relationship between

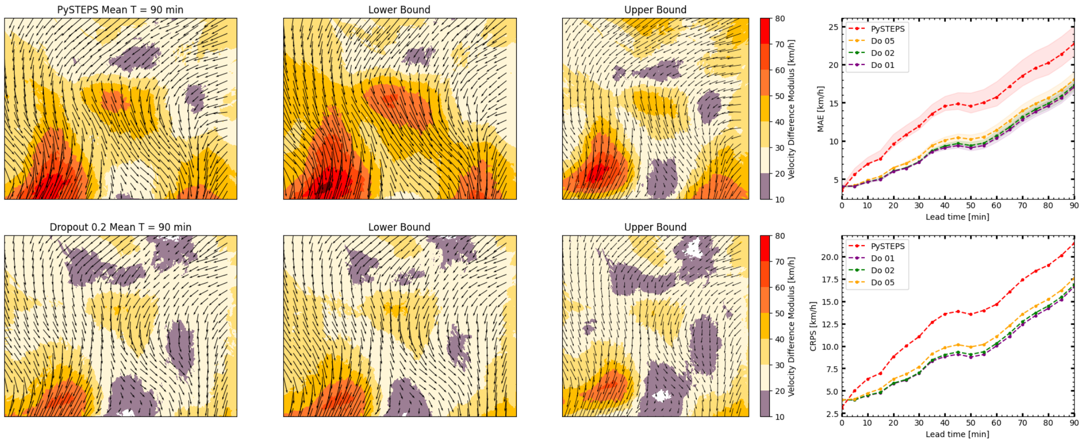


Figure C.6.: Representation of the event of interest from 2021-10-21-0125 related to cyclonic rotations. The top row contains the difference to U_{ref} at 90 min. of the average prediction of the PySTEPS ensemble, the lower and the upper bound. The bottom row is the same for the model with a drop probability of 0.2. On the right, the spatially averaged MAE of those 3 predictions for PySTEPS and dropouts of 0.1, 0.2 and 0.5 are shown together with the average CRPS for each model. The shaded area of the MAE plot is the error gap between the bounds.

the mean and the bounds. Spatially, the characteristics mentioned in the previous examples apply, with bigger changes in PySTEPS but a very similar spatial structure, without unrealistic changes. These examples, with structured and stable convective events, show that the improved predictions of the motion field are the most important factor. Even with a worsening of each member and a smaller spread, the performance is superior. The ensemble generation of PySTEPS for events with such big errors behaves well, similarly to the dropout examples but with a larger spread, which could be interesting to enhance in the new approach.