MS-nowcasting: Operational Precipitation Nowcasting with Convolutional LSTMs at Microsoft Weather

Sylwester Klocek[†], Haiyu Dong[†], Matthew Dixon[¶], Panashe Kanengoni[†], Najeeb Kazmi[†], Pete Luferenko[†,*], Zhongjian Lv[†], Shikhar Sharma[¶], Jonathan Weyn[†] and Siqi Xiang[†]

[†]Microsoft Corporation
[¶]Microsoft Turing
[*]Pete.Luferenko@microsoft.com

Abstract

We present the encoder-forecaster convolutional long short-term memory (LSTM) deep-learning model that powers Microsoft Weather's operational precipitation nowcasting product. This model takes as input a sequence of weather radar mosaics and deterministically predicts future radar reflectivity at lead times up to 6 hours. By stacking a large input receptive field along the feature dimension and conditioning the model's forecaster with predictions from the physics-based High Resolution Rapid Refresh (HRRR) model, we are able to outperform optical flow and HRRR baselines by 20-25% on multiple metrics averaged over all lead times.

1 Motivation

Accurate short-term forecasts ("nowcasts") of precipitation are extremely important for many aspects of people's daily lives, as evidenced by the popularity of services such as Dark Sky. Nowcasting can also identify extreme weather events such as flooding and inform mitigation strategies for these events. With such extreme events predicted to increase in frequency and intensity in a warmer future climate [1], forecasting and early warning will become even more important.

Traditional forecasting methods based on numerical weather prediction (NWP) models, which compute solutions to differential equations governing the physics of the atmosphere, have a few disadvantages when used for nowcasting. Models such as HRRR [2] are computationally expensive to run at high spatial resolution, cannot predict very short-term precipitation very well due to the required spin-up time of the internal physics, and have limited predictive skill beyond a few hours [3]. Methods for predicting short-term precipitation directly from observed weather radar, including extrapolation methods like optical flow and machine learning methods, have emerged as the state-of-the-art for forecasting precipitation within a few hours. [4] introduced the convolutional long short-term memory (ConvLSTM) architecture with good performance for radar-based precipitation nowcasting. Since then, numerous deep learning architectures and benchmark datasets have been used in a research capacity to tackle this problem [e.g. 5–10], including Google Research's MetNet [11], which compares favorably against the physics-based HRRR.

Despite the promising performance of deep-learning-based nowcasting, few providers have implemented such systems for operations. We present MS-nowcasting, a ConvLSTM-based model designed for operational efficiency that now powers Microsoft Weather's nowcasting maps and notifications in the US and Europe. Our model has several notable improvements over prior work in precipitation nowcasting, including a novel method to ingest a large spatial input receptive field, and leveraging physics-based HRRR predictions to condition the model's forecaster component. Our

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2021.

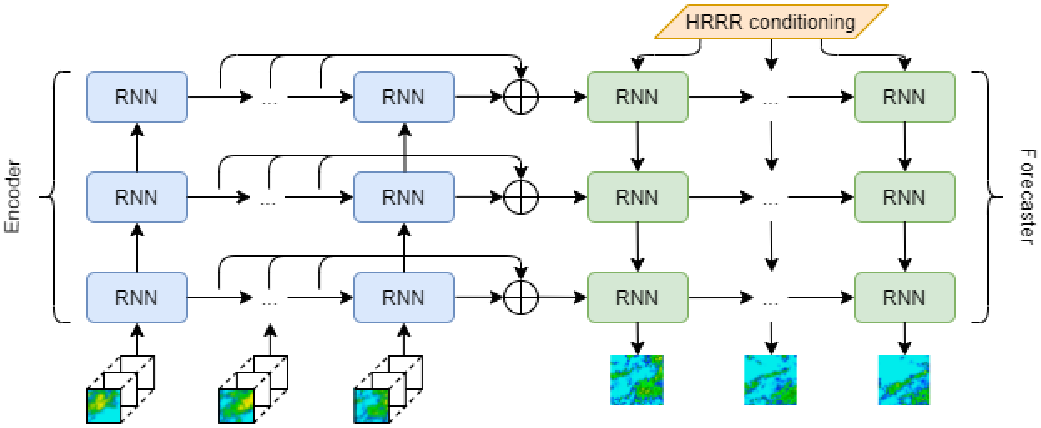

Figure 1: ConvLSTM encoder-forecaster architecture.

design strategies allow MS-nowcasting to run on as little hardware as a single graphics processing unit (GPU). Using only 8 GPUs in production, the model latency for forecasts over the entire US is under two minutes from radar data availability. End-to-end, consumers can enjoy high-quality radar forecast maps within about 6 minutes from the time of radar measurements.

2 Model

To target the goal of delivering the best on the market model accuracy and refresh rate while keeping computational resource requirements to a minimum, we needed to make a few key model design choices. For instance, we needed to avoid expensive operations such as attention mechanism performing $O(n^2)$ operations on tensors. While [11] employed the faster approach of axial attention [12], it necessitated using 256 Tensor Processing Units (TPUs) running in parallel. We decided to leverage existing ConvLSTM architecture first developed in [4] and enhanced in [5], but with some adjustments. Our changes to the original model include adding forecaster conditioning based on an NWP model, adding a large viewport of input spatial information in an atypical manner, tuning hyperparameters, weighting of the hidden states, and optimizer changes.

Our model is a three-layer encoder-forecaster network. The input to the model consists of a sequence of T_i images of weather radar reflectivity (a proxy for precipitation intensity) from the Multi-Radar Multi-Sensor dataset [MRMS, 13], with shape $[B, T_i, C, H_i, W_i]$ and the target is a sequence of T_o radar frames, with shape $[B, T_o, 1, H_o, W_o]$. We perform down convolutions followed by leaky ReLU activations before encoder layers and up convolutions followed by leaky ReLU after forecaster layers similar to [5]. We keep the original ConvLSTM cell instead of methods like TrajGRU and ConvGRU as we found the latter did not provide significant benefits for our problem. Inside this ConvLSTM cell, we perform group normalization after performing the convolutions. This modified version of the ConvLSTM cell is referred to as RNN in Figure 1. Exact hyperparameters of the model architecture can be found in Table 3 in Appendix B.

Nowcasting models and benchmarks often rely on using the same spatial viewport across the entire sequence of both input and predicted target frames. However, over longer prediction sequences and particularly in rapidly changing weather events, additional spatial context in the inputs is necessary to provide information about moving precipitation to the model. MetNet [11] is one example which uses an input viewport larger than the target; specifically, the authors use an input grid of 1024×1024 km to predict a 64×64 km target square, where 1 km corresponds to one image pixel. In order to fit such a large input array into their model, they perform downsampling and cropping. In contrast, we transform our input viewport of 1280×1280 km by dividing the tensor of shape $[1280, 1280]$ into 5 equal segments along both the height and width dimensions and stacking them in the channel dimension to obtain a tensor of shape $[25, 256, 256]$. This large viewport is subsequently referred to as LV. Hence, the input neighbourhood is treated as features by our model. The model's predictions are made for the center 256×256 km square. By extensive search of parameters we found that operating on these spatial dimensions is a good trade-off between speed and forecast quality.

Rather than relying solely on radar data, we also condition our MS-nowcasting model by adding another input to the forecaster cells: the reflectivity forecast product from the HRRR model. By using physics of the atmosphere, HRRR is able to predict formation and dissipation of precipitation within the target frames, not just the evolution of existing precipitation. Our model transforms the input HRRR forecast shape to match the forecaster input shapes. First, we linearly interpolate the tensor in the temporal dimension to the desired number of frames T_o. Next, we perform a series of down convolutions followed by leaky ReLU operations matching those of the encoder layers. The result is a tensor of shape $[B, T_o, 1, H_{l3}, W_{l3}]$, which can be concatenated directly to the state in the third layer of the forecaster.

Unlike the previous ConvLSTM publications [4, 5], we use all hidden encoder ConvLSTM states to produce forecaster hidden states. For this purpose, we introduce a vector of trainable weights $w = \langle w_1, \ldots, w_m \rangle$, and each weight has a length of m corresponding to temporal dimension of MRMS input. The forecaster hidden state h_l with shape $[B, 1, C_l, H_l, W_l]$, where l is the layer (up to 3), is computed as

$$h_l = \sum_{i=1}^{m} w_i \cdot h_{B,t_i,C_l,H_l,W_l} \tag{1}$$

This operation is represented as the $\bigoplus$ symbol in Figure 1. Using more than one past hidden state of the encoder resulted in slight metrics improvements. The cell state of the last encoder RNN cell remains unchanged.

For training purposes, we use the average of mean absolute error (MAE) and mean squared error (MSE) as the loss function, where each frame is weighted by the B-MAE thresholds borrowed from [5]. We use Adam optimizer [14] with stochastic weight averaging [SWA, 15] on top - this helped our model converge to wider optima, generalize better, and train more stably through mode connectivity [16]. We set the learning rate to 2×10^{-4}, gradient clipping to 1.0 and weight decay to 1×10^{-4}. For speed and multi-GPU environment training environment, we use the DeepSpeed framework [17].

3 Experiments

We compare our model to persistence, a naive baseline whereby weather is forecast to remain unchanged, an optical flow method implemented in the *rainymotion* library described in [18], and the radar reflectivity forecast from HRRR.

We perform four ablation experiments to illustrate the improvements achieved with our adjustments to the existing ConvLSTM [4, 5].

1. **MS-nowcasting**: baseline model using a small 256 × 256 km input viewport only.
2. **MS-nowcasting + HRRR**: MS-nowcasting with HRRR input to the forecaster.
3. **MS-nowcasting + LV**: MS-nowcasting with LV and no HRRR input to the forecaster.
4. **MS-nowcasting + HRRR + LV**: MS-nowcasting with both LV and HRRR input to the forecaster.

3.1 Data

We prepared the training data by sampling 83,000 sequences of at least 440 minutes from the MRMS archive for 2018, from which we sample 10 million 440-minute windows. The first 80 minutes of these windows are model inputs and the next 360 minutes are targets. In order to reduce compute costs in both training and inference, we use 20 input frames at a 4-minute resolution and 45 target frames at an 8-minute resolution. For the test and validation sets, we drew 1500 and 500 windows, respectively, from the MRMS archive for 2019.

3.2 Results

We evaluate the mean absolute error (MAE) and F1 score at the threshold of 12 dBZ (corresponding to precipitation rates of approximately 0.2 mm/h) of the predictions against ground truth. We also evaluate the Multiscale Structural Similarity Index Measure (MS-SSIM) and Peak Signal-to-Noise Ratio (PSNR) to measure image quality relative to ground truth. Table 1 shows that our model

Table 1: Metrics averaged over lead times 0–2 hours

Model	MAE	F1 (12 dBZ)	MS-SSIM	PSNR
Persistence	4.88	0.686	0.5156	21.21
Optical Flow	4.39	0.720	0.5527	21.83
HRRR	5.54	0.610	0.4634	20.01
MS-nowcasting	3.64	0.788	0.6419	23.23
+ HRRR	3.55	0.780	0.6510	23.29
+ LV	3.28	0.808	0.6678	23.86
+ HRRR + LV	**3.23**	**0.813**	**0.6721**	**24.03**

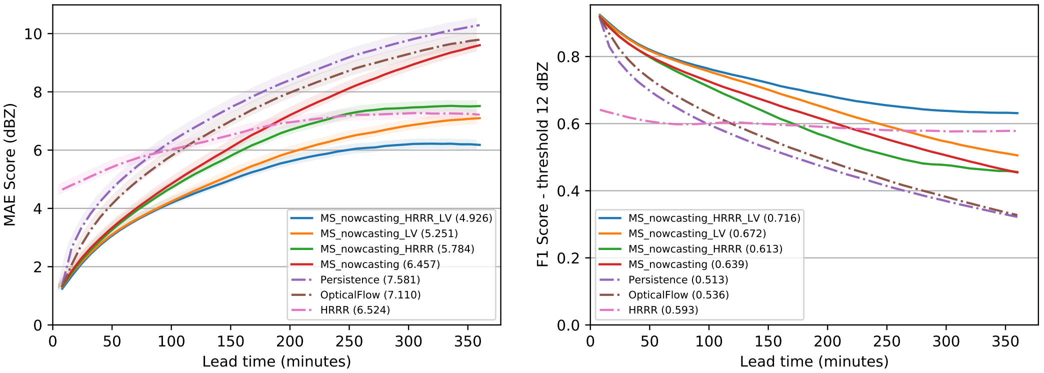

Figure 2: MAE and F1 metrics over lead times for the ablation study. Numbers in parentheses in the plot legends are the average metric value over all lead times.

(MS-nowcasting + HRRR + LV) substantially outperforms all others on all metrics in the first two hours, while even MS-nowcasting is better than all three baselines in the same period.

Figure 2 shows the metrics by lead time for the ablation study. While MS-nowcasting starts off better than HRRR, its performance becomes worse at longer lead times. Adding HRRR input improves performance on MAE, but worsens it on F1. This is explained by the model bias presented in Figure 3 in Appendix A. MS-nowcasting has a high overprediction bias, especially at long lead times. Conversely, adding HRRR leads to a high underprediction bias, which causes the model to miss the 12 dBZ threshold more frequently. On the other hand, adding LV alone gives more gains than adding HRRR alone as that model has less of an underprediction bias.

The final model with both LV and HRRR outperforms all the rest at all lead times, demonstrating that both larger spatial context and formation and dissipation of precipitation provided by HRRR are important in predicting longer sequences of precipitation. Nevertheless, the choice of MAE+MSE loss for the model results in predictions that appear overly smooth (see Figs. 4 and 5 in the Appendix). Recent studies have shown that generative adversarial networks are a potential way of addressing this deficiency [10].

4 Conclusion

In this paper, we presented MS-nowcasting, an encoder-forecaster long short-term memory (LSTM) deep-learning model for precipitation nowcasting. MS-nowcasting improves upon the existing ConvLSTM [4, 5] architecture and outperforms the operational HRRR NWP model for lead times of up to 6 hours. Our approach allows for efficient and accurate rapid-update precipitation nowcasting. The model is operational and can be potentially inform forecasts for early warning of extreme weather events.

Acknowledgments and Disclosure of Funding

The authors thank the many engineers at Microsoft who helped make the production model a reality.

References

[1] Lorenzo Alfieri, Berny Bisselink, Francesco Dottori, Gustavo Naumann, Ad de Roo, Peter Salamon, Klaus Wyser, and Luc Feyen. Global projections of river flood risk in a warmer world. *Earth's Future*, 5(2):171–182, 2017.

[2] Stanley G Benjamin, Stephen S Weygandt, John M Brown, Ming Hu, Curtis R Alexander, Tatiana G Smirnova, Joseph B Olson, Eric P James, David C Dowell, Georg A Grell, et al. A north american hourly assimilation and model forecast cycle: The rapid refresh. *Monthly Weather Review*, 144(4):1669–1694, 2016.

[3] Dale R Durran and Jonathan A Weyn. Thunderstorms do not get butterflies. *Bulletin of the American Meteorological Society*, 97(2):237–243, 2016.

[4] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[5] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and A new model. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5617–5627, 2017.

[6] Vadim Lebedev, Vladimir Ivashkin, Irina Rudenko, Alexander Ganshin, Alexander Molchanov, Sergey Ovcharenko, Ruslan Grokhovetskiy, Ivan Bushmarinov, and Dmitry Solomentsev. Precipitation nowcasting with satellite imagery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2680–2688, 2019.

[7] Georgy Ayzel, Tobias Scheffer, and Maik Heistermann. Rainnet v1. 0: a convolutional neural network for radar-based precipitation nowcasting. *Geoscientific Model Development*, 13(6):2631–2644, 2020.

[8] Lei Chen, Yuan Cao, Leiming Ma, and Junping Zhang. A deep learning-based methodology for precipitation nowcasting with radar. *Earth and Space Science*, 7(2), 2020.

[9] Mark Veillette, Siddharth Samsi, and Chris Mattioli. Sevir: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in Neural Information Processing Systems*, 33, 2020.

[10] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, Rachel Prudden, Amol Mandhane, Aidan Clark, Andrew Brock, Karen Simonyan, Raia Hadsell, Niall Robinson, Ellen Clancy, Alberto Arribas, and Shakir Mohamed. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, Sep 2021.

[11] Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. Metnet: A neural weather model for precipitation forecasting. *CoRR*, abs/2003.12140, 2020.

[12] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *CoRR*, abs/1912.12180, 2019.

[13] Jian Zhang, Kenneth Howard, Carrie Langston, Brian Kaney, Youcun Qi, Lin Tang, Heather Grams, Yadong Wang, Stephen Cocks, Steven Martinaitis, et al. Multi-radar multi-sensor (mrms) quantitative precipitation estimation: Initial operating capabilities. *Bulletin of the American Meteorological Society*, 97(4):621–638, 2016.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 876–885. AUAI Press, 2018.

[16] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8803–8812, 2018.

[17] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3505–3506. ACM, 2020.

[18] Georgy Ayzel, Maik Heistermann, and Tanja Winterrath. Optical flow models as an open benchmark for radar-based precipitation nowcasting (rainymotion v0.1). *Geoscientific Model Development*, 12(4):1387–1402, 2019.

A Appendix: Additional metrics

Table 2: Metrics averaged over all lead times 0–6 hours

Model	MAE	F1 (12 dBZ)	MS-SSIM	PSNR
Persistence	7.58	0.513	0.3626	16.98
Optical Flow	7.11	0.536	0.3780	17,38
HRRR	6.52	0.593	0.3905	17.72
MS-nowcasting	6.46	0.639	0.4788	18.78
+ HRRR	5.78	0.672	0.4952	19.10
+ LV	5.25	0.613	0.5137	19.68
+ HRRR + LV	**4.92**	**0.716**	**0.5291**	**20.16**

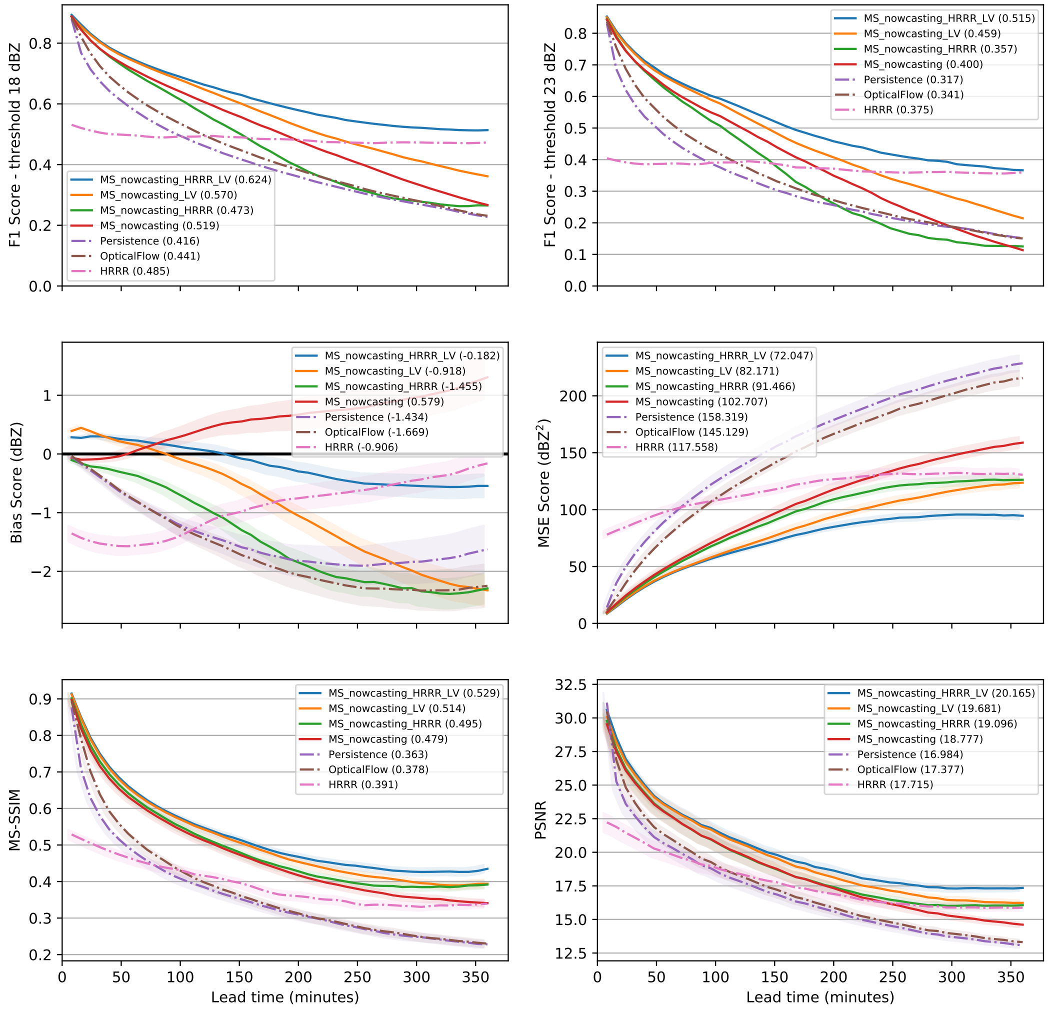

Figure 3: As in Fig. 2, but including more metrics (as labeled). F1 thresholds of 18 and 23 dBZ correspond to precipitation rates of roughly 1.0 and 2.0 mm/h respectively.

B Appendix: Architecture

Table 3: Model architecture

Name	Kernel size	Size	Stride	Padding
hidden state weights	-	[20]	-	-
L0-encoder-downsconv-weight	6x6	[16, 25, 6, 6]	3	0
L0-encoder-downsconv-bias	-	16	-	-
L0-encoder convlstmcell i_g, f_g, c_g, o_g	3x3	[256, 80, 3, 3]	1	1
L0-encoder convlstmcell biases	-	[256]	-	-
L0-encoder groupnorm weight	-	[256]	-	-
L0-encoder groupnorm bias	-	[256]	-	-
L1-encoder-downsconv-weight	5x5	[192, 64, 5, 5]	3	1
L1-encoder-downsconv-bias	-	[192]	-	-
L1-encoder convlstmcell i_g, f_g, c_g, o_g	3x3	[768, 384, 3, 3]	1	1
L1-encoder convlstmcell biases	-	[768]	-	-
L1-encoder groupnorm weight	-	[768]	-	-
L1-encoder groupnorm bias	-	[768]	-	-
L2-encoder-downsconv-weight	3x3	[192, 192, 3, 3]	2	1
L2-encoder-downsconv-bias	-	[192]	-	-
L2-encoder convlstmcell i_g, f_g, c_g, o_g	3x3	[768, 384, 3, 3]	1	1
L2-encoder convlstmcell biases	-	[768]	-	-
L2-encoder groupnorm weight	-	[768]	-	-
L2-encoder groupnorm bias	-	[768]	-	-
HRRR conditioning downconv-0 weight	6x6	[16, 1, 6, 6]	3	0
HRRR conditioning downconv-0 bias	-	[16]	-	-
HRRR conditioning downconv-1 weight	6x6	[192, 16, 5, 5]	3	1
HRRR conditioning downconv-1 bias	-	[192]	-	-
HRRR conditioning downconv-2 weight	3x3	[192, 192, 3, 3]	2	1
HRRR conditioning downconv-2 bias	-	[192]	-	-
L2-forecaster-convlstmcell i_g, f_g, c_g, o_g	3x3	[768, 384, 3, 3]	1	1
L2-forecaster-convlstmcell bias	-	[768]	-	-
L2-forecaster groupnorm weight	-	[768]	-	-
L2-forecaster groupnorm biases	-	[768]	-	-
L2-forecaster upconv weight	4x4	[192, 192, 4, 4]	2	1
L2-forecaster upconv bias	-	[192]	-	-
L1-forecaster-convlstmcell i_g, f_g, c_g, o_g	3x3	[768, 384, 3, 3]	1	1
L1-forecaster-convlstmcell bias	-	[768]	-	-
L1-forecaster groupnorm weight	-	[768]	-	-
L1-forecaster groupnorm biases	-	[768]	-	-
L1-forecaster upconv weight	5x5	[192, 64, 5, 5]	3	1
L1-forecaster upconv bias	-	[64]	-	-
L0-forecaster-convlstmcell i_g, f_g, c_g, o_g	3x3	[256, 128, 3, 3]	1	1
L0-forecaster-convlstmcell bias	-	[256]	-	-
L0-forecaster groupnorm weight	-	[256]	-	-
L0-forecaster groupnorm biases	-	[256]	-	-
L0-forecaster upconv weight	7x7	[64, 16, 7, 7]	3	0
L0-forecaster upconv bias	-	[16]	-	-
final-conv.0.weight	3x3	[16, 16, 3, 3]	1	1
final-conv.0.bias	-	[16]	-	-
final-conv.2.weight	1x1	[1, 16, 1, 1]	1	0
final-conv.2.bias	-	[1]	-	-

C Appendix: Sample predictions

Figures 4 and 5 show two example sequences of predicted and observed true radar reflectivity from the test set. In both cases, the precipitation evolves substantially over the 6-hour forecast period. In Fig. 4, a new line of thunderstorms develops behind (to the northwest of) the original band of precipitation and intensifies. The HRRR model generally captures this evolution but produces too scattered and weak thunderstorms. Our best MS-nowcasting + HRRR + LV captures the general behavior of the precipitation pattern but its forecasts are too smooth and weak. Meanwhile, the other MS-nowcasting models without the LV and/or HRRR conditioning greatly dissipate the rain, while the raw HRRR predicts precipitation in incorrect locations.

In Fig. 5, rain is progressing from west to east into the target region while also intensifying. The optical flow baseline, despite containing the full 1280×1280 input, fails to capture any precipitation, as does the MS-nowcasting model with no large viewport. Meanwhile the MS-nowcasting + HRRR + LV model accurately picks up on the large area of precipitation, although it loses most of the detailed structure.

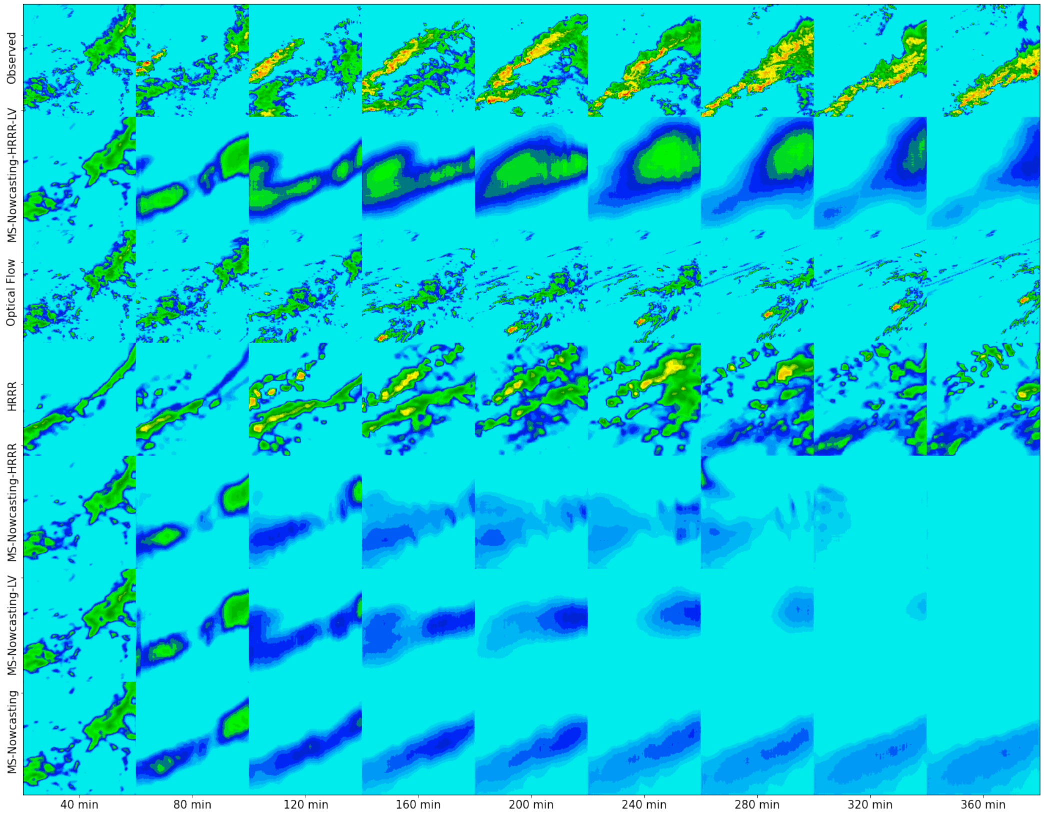

Figure 4: An example sequence of observed true radar reflectivity (top row) and predicted radar from our models and the optical flow and HRRR baselines, as labeled.

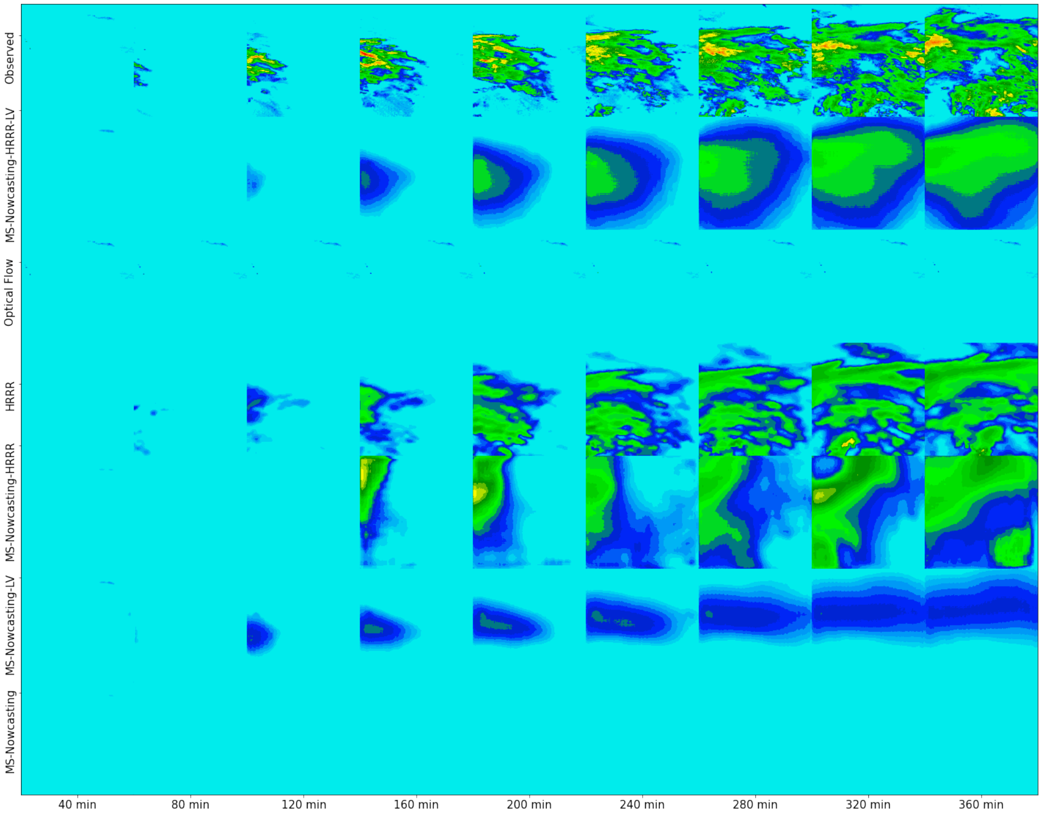

Figure 5: As in Fig. 4, but a different example sequence.

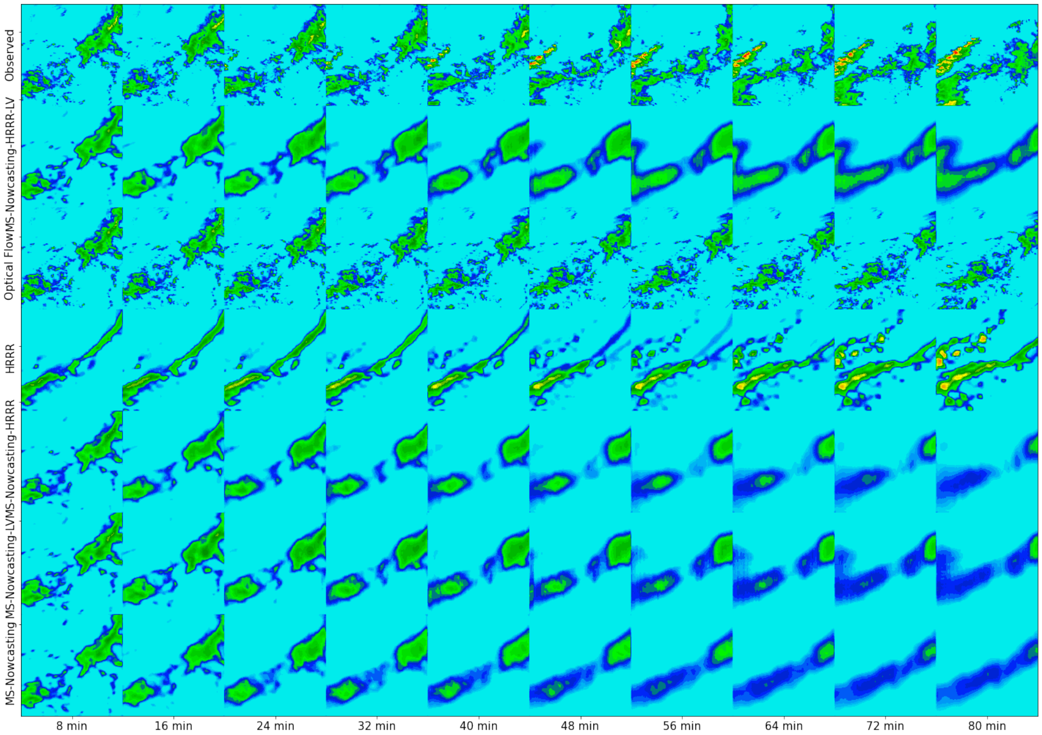

Figure 6: An in Fig . 4, but predictions for 80 mins into the future with 8 min interval.