BlockGPT: Spatio-Temporal Modelling of Rainfall via Frame-Level Autoregression

Cristian Meo[*]
LatentWorlds AI
TUDelft
c.meo@tudelft.nl
Netherlands

Varun Sarathchandran[*]
TUDelft
Netherlands
v.sarathchandran@tudelft.nl

Avijit Majhi[*]
TUDelft
Netherlands
a.majhi@tudelft.nl

Shao-Hsuan Hung
TUDelft
Netherlands
shaohung@tudelft.nl

Carlo Saccardi
TUDelft
Netherlands

Ruben Imhoff
Deltares
Netherlands

Roberto Deidda
University of Cagliari
Italy

Remko Uijlenhoet
TUDelft
Netherlands

Justin Dauwels
TUDelft
Netherlands

Abstract

Predicting precipitation maps is a highly complex spatiotemporal modeling task, critical for mitigating the impacts of extreme weather events. Short-term precipitation forecasting, or nowcasting, requires models that are not only accurate but also computationally efficient for real-time applications. Current methods, such as token-based autoregressive models, often suffer from flawed inductive biases and slow inference, while diffusion models can be computationally intensive. To address these limitations, we introduce BlockGPT, a generative autoregressive transformer using batched tokenization (Block) method that predicts full two-dimensional fields (frames) at each time step. Conceived as a model-agnostic paradigm for video prediction, BlockGPT factorizes space–time by using self-attention within each frame and causal attention across frames; in this work, we instantiate it for precipitation nowcasting. We evaluate BlockGPT on two precipitation datasets, viz. KNMI (Netherlands) and SEVIR (U.S.), comparing it to state-of-the-art baselines including token-based (NowcastingGPT) and diffusion-based (DiffCast+Phydnet) models. The results show that BlockGPT achieves superior accuracy, event localization as measured by categorical metrics, and inference speeds up to $31\times$ faster than comparable baselines. Here there is the official implemention of BlockGPT:https://github.com/Cmeo97/BlockGPT.

1 Introduction

Climate change is increasing the frequency and intensity of extreme rainfall worldwide, disrupting infrastructure and posing risks to life and property Alfieri et al. [2017], Martinkova and Kysely [2020], Klocek et al. [2021], Czibula et al. [2021], Malkin Ondík et al. [2022]. This amplifies the need for accurate, high-resolution short-term weather forecasting (nowcasting) Côté et al. [2015]. In operational early-warning chains Imhoff et al. [2020, 2023], short-range forecasts are typically produced by numerical weather prediction (NWP) systems; however, for minute-to-hour lead times, practical constraints—latency, update frequency, and effective resolution—can hinder the timing

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2025.

and localization of intense rainfall Bauer et al. [2015], Berenguer et al. [2012], Pierce et al. [2012]. Crucially, NWP and radar-based nowcasting are complementary: NWPs provide large-scale dynamical context and longer lead times, whereas radar nowcasting leverages high-resolution observations for within-event, local decision-making. Rainfall nowcasting classically denotes statistical/heuristic extrapolation of real-time quantitative precipitation estimates (QPEs), exploiting radar's fine spatiotemporal resolution (often $\sim$1 km/5 min) and direct initialization from the latest observations Overeem et al. [2009]. Methods include (i) field-based advection with stochastic evolution Seed [2003], Bowler et al. [2006], Berenguer et al. [2011], Seed et al. [2013], Sokol et al. [2017], Ayzel et al. [2019], (ii) object-oriented cell tracking Dixon and Wiener [1993], Han et al. [2009], (iii) analogue approaches Atencia and Zawadzki [2014, 2015], and (iv) machine learning Shi et al. [2015], Ravuri et al. [2021], Luo et al. [2021], Liu et al. [2022]. Community efforts such as `pysteps` have consolidated and advanced these approaches in open source Pulkkinen et al. [2019]. Recent work reframes radar nowcasting as a video prediction task, learning to propagate precipitation fields over minute-to-multi-hour horizons with low latency Shi et al. [2015], Ravuri et al. [2021], Prudden et al. [2020]. In practice, data-driven nowcasts guide local decisions at short leads ($\approx$0–3 h), while NWP supplies the large-scale dynamics for longer horizons and basin- to synoptic-scale planning Bauer et al. [2015]. This perspective motivates modern generative sequence models for video prediction. State-of-the-art (SOTA) approaches employ VQ-VAEs Van Den Oord et al. [2017], Meo et al. [2024a], transformers Vaswani et al. [2017], Meo et al. [2024b], Bi et al. [2023], Yin et al. [2024], and diffusion models Gao et al. [2023], Yu et al. [2024] to improve accuracy and efficiency. Despite progress, long-term consistency, latency, and computational cost remain key challenges. To address these limitations, we introduce BlockGPT, a transformer that models spatiotemporal rainfall dynamics via frame-level autoregression. By predicting entire precipitation fields at each step, BlockGPT avoids the inductive biases and computational bottlenecks of token-level autoregression, yielding more coherent predictions and faster inference. Our contributions are: (1) a generative transformer that autoregressively predicts full precipitation fields, contrasting with prior token-level approaches; (2) a comprehensive evaluation on SEVIR Veillette et al. [2020] and KNMI Overeem and Imhoff [2020] showing state-of-the-art categorical skill and event localization, with inference up to 31$\times$ faster than SOTA baselines.

2 Related Work

Early deep learning efforts for nowcasting relied on recurrent neural networks (RNNs) Rumelhart et al. [1986] architectures to model temporal sequences. Models like ConvLSTM Shi et al. [2015] and ConvGRU Shi et al. [2017] adapted RNNs designed for spatio-temporal data by replacing matrix multiplications with convolutional operations. This line of work was extended by models such as TrajGRU Shi et al. [2017], which improved motion tracking, and PhyDNet Guen and Thome [2020], which embedded physical constraints by decomposing the latent space. DGMR Ravuri et al. [2021] employed a Generative Adversarial Network (GAN) Goodfellow et al. [2014] with spatial and temporal discriminators to improve forecast quality. More recently, diffusion models Ho et al. [2020] have become prominent because of their stable training and high-fidelity generation. Models like PreDiff Gao et al. [2023] perform denoising in a latent space to generate future frames. However, it requires more than 30 days to be trained. In contrast, the proposed BlockGPT can be trained in 1.5 hours. DiffCast+Phydnet Yu et al. [2024], a key baseline in our work, introduced a residual diffusion approach where a base model predicts a coarse forecast and a diffusion model learns to predict the stochastic residual. Transformers offer an alternative to recurrent models that has been shown to be more stable, efficient, and scalable, leveraging self-attention to capture long-range dependencies Vaswani et al. [2017], Meo et al. [2025]. MAU Chang et al. [2021], for example, integrates motion cues through temporal aggregation, while Earthformer Gao et al. [2022] applies cuboidal self-attention over radar volumes. Most closely related to our work is NowcastingGPT Meo et al. [2024b], which tokenizes radar precipitation fields using a VQ-VAE Van Den Oord et al. [2017] and autoregressively predicts them with a transformer decoder. However, the autoregression in NowcastingGPT operates at the token level rather than in time, creating an ill-posed learning problem that results in fragmented outputs and slow inference. To address these limitations, we propose BlockGPT, a generative transformer model that predicts entire precipitation fields at once in latent space, employing a block attention mask to enable bidirectional spatial attention within each precipitation field while maintaining temporal causality across precipitation fields. Motivated readers can also find a releted work section about video prediction architectures in appendix D.1.

3 Methodology: BlockGPT Pipeline

Given a sequence of T_c context precipitation fields $X_{1:T_c}$, with $X_t \in \mathbb{R}^{H \times W}$, where $H \times W$ is the grid size of the rainfall fields, and the task is to predict the following T future precipitation fields. The proposed BlockGPT pipeline, decomposes the prediction task into two stages: (1) compressing precipitation fields into a latent token space, and (2) autoregressively modelling temporal dynamics.

Feature Exctraction Each precipitation field X_t is encoded using a VQ-GAN Esser et al. [2021], which downsamples and discretises it into a grid of latent tokens $\boldsymbol{\mathcal{T}}_t \in \mathbb{R}^{H' \times W'}$. A detailed description of the employed VQ-GAN and its training procedure can be found in appendix C.

Dynamics Modeling Prior work such as NowcastingGPT Meo et al. [2024b] models temporal evolution by flattening each grid $\boldsymbol{\mathcal{T}}_t$ into a 1D token sequence $\mathbf{z}_t$, and concatenating across time $\mathbf{z}_t = \left(z_t^{(1)}, z_t^{(2)}, \ldots, z_t^{(H'W')}\right)$, with the joint distribution factorised autoregressively as:

$$p(\mathbf{z}) = \prod_{i=1}^{T \cdot H'W'} p\left(z^{(i)} \mid z^{(1)}, z^{(2)}, \ldots, z^{(i-1)}\right). \tag{1}$$

However, such formulation implicitly assumes a sequential correlation of all $z^{(t)}$, imposing a flawed inductive bias. Indeed, the spatial tokens $z^{(t)}$ within a precipitation field are bidirectionally correlated and not naturally sequential. Treating them as a causal chain forces the model to predict inherently co-dependent tokens autoregressively, which is an ill-posed modelling assumption and leads to inefficient, suboptimal decoding. By contrast, we factorise the joint distribution as:

$$p(\boldsymbol{\mathcal{T}}) = \prod_{t=1}^{T} p\left(\boldsymbol{\mathcal{T}}_t \mid \boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_{t-1}\right), \tag{2}$$

where each $\boldsymbol{\mathcal{T}}_t$ is holistically modelled, preserving the original 2D structure that contains bidirectionally correlated features. This distribution shift has several advantages, the first and most important is fixing the flawed inductive bias - features are not anymore assumed to be sequentially correlated. Secondly, the autoregression step is performed at the actual time scale, which allows the model to learn a meaningful time dependent dynamics. Finally, inference is $H'W'$ times faster by design, since we can now infer a complete $\boldsymbol{\mathcal{T}}_t$ with a single forward pass. It is important to note that, during training and inference, we use block attention masks - spatial tokens within a precipitation field are allowed to attend bidirectionally, while temporal attention is strictly causal. This design more naturally aligns with the spatiotemporal structure of precipitation: radar maps require full-field spatial modeling, whereas future precipitation fields should depend only on the past context.

4 Experiments

We evaluate all models on the task of nowcasting, where the goal is to predict the next 6 radar precipitation fields given 3 context precipitation fields. Each precipitation field represents 30 minutes of precipitation, resulting in a forecast horizon of 3 hours. Experiments are conducted on two real-world radar datasets: the Dutch KNMI dataset Overeem and Imhoff [2020] and the SEVIR dataset Veillette et al. [2020] from the United States. Appendix A presents analyses of the considered datasets, providing a detailed overview of the datasates statistics. We benchmark BlockGPT against NowcastingGPT Meo et al. [2024b] and DiffCast+Phydnet Yu et al. [2024]. To the best of our knowledge, the former is the current state-of-the-art discrete token-based autoregressive model for precipitation nowcasting in the KNMI dataset, while the latter exemplifies the residual diffusion paradigm Yu et al. [2024] and is the state-of-the-art in the SEVIR dataset. We report quantitative results in terms of four key metrics: Mean Squared Error (MSE), Pearson Correlation Coefficient (PCC), Critical Success Index (CSI), and False Alarm Rate (FAR). Details about the experiments can be found in Appendix E.

4.1 Results

In this section, we design empirical experiments to understand the performance of BlockGPT and its potential limitations by exploring the following questions: (1) How does BlockGPT's frame-level autoregressive approach compare to token-level autoregression (NowcastingGPT) and diffusion-based models (DiffCast+Phydnet) in terms of prediction accuracy and computational efficiency? (2) What differences are there between autoregressive and diffusion-based generative behaviors?

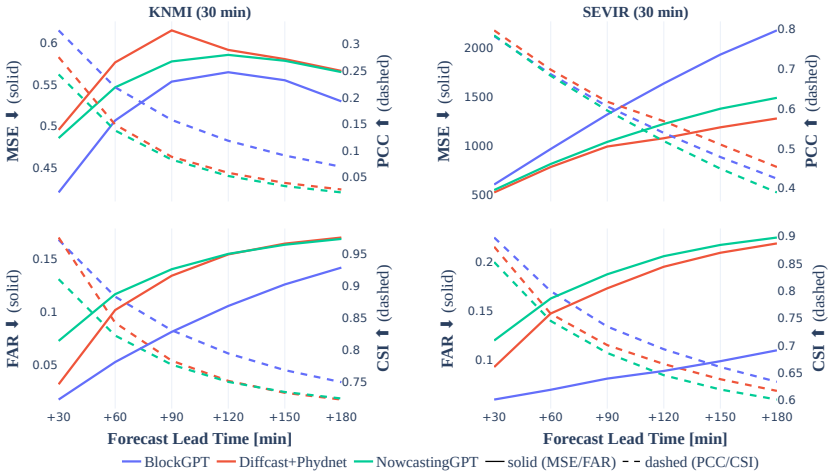

Figure 1: MSE, PCC, CSI, and FAR of BlockGPT and related baselines, on KNMI and SEVIR datasets. Results are averaged across 3 seeds.

Block Autoregressive models are better nowcasters than diffusion models. Figure 1 presents a comparative performance analysis of three models: Diffcast+Phydnet, BlockGPT, and NowcastingGPT on the KNMI and SEVIR datasets for forecast lead times up to 180 minutes. A consistent performance hierarchy is evident across both datasets, all forecast lead times and all metrics, with BlockGPT outperforming Diffcast+Phydnet and NowcastingGPT, with the only exception for MSE and PCC evaluation on SEVIR where DiffCast+Phydnet performs the best (Diffcast + Phydnet: $\sim$ 1250 $\rightarrow$ NowcastingGPT: $\sim$ 1450 $\rightarrow$ BlockGPT: $\sim$ 2100). However, BlockGPT shows superior event-level detection performance as reflected in higher CSI and lower FAR values. Qualitative results can be found in Appendix B.1 and the percentile-wise continuous metrics in Appendix B.2. As illustrated by the qualitative case studies in Appendix B.1, BlockGPT preserves storm morphology and displacement more faithfully than the baselines across KNMI and SEVIR events (see Fig. 4, 5, 6, 7). On SEVIR, BlockGPT occasionally overestimates high intensity storm cells at longer lead times—consistent with the continuous scores reported in App. B.2 (see Fig. 13), whereas on KNMI, errors decrease toward over the leadtimes and PCC remains systematically higher (see Fig. 12).
BlockGPT is more robust than diffusion pipelines. To further evaluate model performance at various rainfall intensities, we also report the Area Under the ROC Curve (AUC) over time for different precipitation thresholds on the KNMI dataset in Fig. 8. BlockGPT consistently outperforms both baselines at all thresholds and time steps, demonstrating robustness in detecting precipitation events of varying severity. Consistent trends are observed at the catchment scale in B.3, where AUC-ROC computed for 1, 2, and 8 mm h^{-1} across +30 to +180 min lead times confirms that BlockGPT maintains the highest detection skill across thresholds (Figure 8).

5 Conclusion

In this work, we introduced BlockGPT, a frame-level autoregressive transformer designed for precipitation nowcasting. By shifting the generative process from a token-by-token sequence to predicting entire precipitation fields autoregressively, BlockGPT overcomes the flawed inductive biases and computational bottlenecks of prior token-level models, resulting in a 31x faster inference speed than its counterparts, as showed in Appendix G.2 and F. This approach enables the use of bidirectional attention to capture complex spatial patterns within each frame while strictly maintaining temporal causality across frames. Our comprehensive evaluation on the KNMI and SEVIR datasets demonstrates that BlockGPT consistently outperforms state-of-the-art models on key categorical metrics, including the CSI and Area Under the ROC Curve. This indicates a stronger ability to accurately localize and predict precipitation events, particularly those exceeding critical intensity thresholds. Future work could focus on enhancing fine-grained prediction accuracy, potentially by integrating BlockGPT as a powerful backbone within a residual diffusion framework. Further research could also explore the incorporation of physical constraints and the development of robust uncertainty quantification methods to improve prediction reliability for critical decision-making.

Acknowledgments

This work is part of the project Delft AI4WF: Delft Artificial Intelligence for Weather Forecast with file number 2024.023 of the research programme Computing time of national computer systems, which is (partly) financed by the NWO under the grant https://doi.org/10.61686/VYGRS56933.

References

Lorenzo Alfieri, B Bisselink, Francesco Dottori, Gustavo Naumann, A. P. J. De Roo, Péter Salamon, Klaus Wyser, and Luc Feyen. Global projections of river flood risk in a warmer world. *Earth's Future*, 5, 2017. URL https://api.semanticscholar.org/CorpusID:42772267.

A. Atencia and I. Zawadzki. A comparison of two techniques for generating nowcasting ensembles. part i: Lagrangian ensemble technique. *Monthly Weather Review*, 142:4036–4052, 2014. doi: 10.1175/MWR-D-13-00117.1.

Aitor Atencia and Isztar Zawadzki. A comparison of two techniques for generating nowcasting ensembles. part ii: Analogs selection and comparison of techniques. *Monthly Weather Review*, 143(7):2890–2908, 2015.

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.

Cong Bai, Feng Sun, Jinglin Zhang, Yi Song, and Shengyong Chen. Rainformer: Features extraction balanced network for radar-based precipitation nowcasting. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. URL https://api.semanticscholar.org/CorpusID:248132089.

Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

M. Berenguer, M. Surcel, I. Zawadzki, M. Xue, and F. Kong. The diurnal cycle of precipitation from continental radar mosaics and numerical weather prediction models. part ii: Intercomparison among numerical models and with nowcasting. *Monthly Weather Review*, 140:2689–2705, 2012. doi: 10.1175/MWR-D-11-00181.1.

Marc Berenguer, Daniel Sempere-Torres, and Geoffrey G.S. Pegram. Sbmcast – an ensemble nowcasting technique to assess the uncertainty in rainfall forecasts by lagrangian extrapolation. *Journal of Hydrology*, 404(3):226–240, 2011. ISSN 0022-1694. doi: 10.1016/j.jhydrol.2011.04.033.

Haoran Bi, Maksym Kyryliuk, Zhiyi Wang, Cristian Meo, Yanbo Wang, Ruben Imhoff, Remko Uijlenhoet, and Justin Dauwels. Nowcasting of extreme precipitation using deep generative models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10094988.

Neill E Bowler, Clive E Pierce, and Alan W Seed. Steps: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled nwp. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 132(620):2127–2155, 2006.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. Mau: A motion-aware unit for video prediction and beyond. *Advances in Neural Information Processing Systems*, 34:26950–26962, 2021.

J Côté, C Jablonowski, P Bauer, and N Wedi. Seamless prediction of the earth system: From minutes to months, 2015.

Gabriela Serban Czibula, Andrei Mihai, Alexandra-Ioana Albu, István Gergely Czibula, Sorin Burcea, and Abdelkader Mezghani. Autonowp: An approach using deep autoencoders for precipitation nowcasting based on weather radar reflectivity prediction. *Mathematics*, 2021. URL https://api.semanticscholar.org/CorpusID:238018677.

M. Dixon and G. Wiener. TITAN: Thunderstorm identification, tracking, analysis, and nowcasting—a radar-based methodology. *Journal of Atmospheric and Oceanic Technology*, 10:785–797, 1993.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. *ArXiv*, abs/2207.05833: 25390–25403, 2022. URL https://api.semanticscholar.org/CorpusID:250492774.

Zhihan Gao, Xingjian Shi, Boran Han, Hongya Wang, Xiaoyong Jin, Danielle C. Maddix, Yi Zhu, Mu Li, and Bernie Wang. Prediff: Precipitation nowcasting with latent diffusion models. *ArXiv*, abs/2307.10422:78621–78656, 2023. URL https://api.semanticscholar.org/CorpusID:259991562.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Steven J Goodman, Richard J Blakeslee, William J Koshak, Douglas Mach, Jeffrey Bailey, Dennis Buechler, Larry Carey, Chris Schultz, Monte Bateman, Eugene McCaul Jr, et al. The goes-r geostationary lightning mapper (glm). *Atmospheric research*, 125:34–49, 2013.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020.

L. Han, S. Fu, L. Zhao, Y. Zheng, H. Wang, and Y. Lin. 3d convective storm identification, tracking, and forecasting—an enhanced TITAN algorithm. *Journal of Atmospheric and Oceanic Technology*, 26:719–732, 2009. doi: 10.1175/2008JTECHA1084.1.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

R. O. Imhoff, C. C. Brauer, A. Overeem, A. H. Weerts, and R. Uijlenhoet. Spatial and temporal evaluation of radar rainfall nowcasting techniques on 1,533 events. *Water Resources Research*, 56(8):e2019WR026723, 2020. doi: https://doi.org/10.1029/2019WR026723. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR026723. e2019WR026723 10.1029/2019WR026723.

R. O. Imhoff, L. De Cruz, W. Dewettinck, C. C. Brauer, R. Uijlenhoet, K.-J. van Heeringen, et al. Scale-dependent blending of ensemble rainfall nowcasts and numerical weather prediction in the open-source pysteps library. *Quarterly Journal of the Royal Meteorological Society*, 149(753): 1335–1364, 2023. doi: 10.1002/qj.4461.

Ian T. Jolliffe and David B. Stephenson, editors. *Forecast Verification: A Practitioner's Guide in Atmospheric Science*. John Wiley & Sons, Chichester, UK, 2 edition, 2003. ISBN 9780470864418.

Sylwester Klocek, Haiyu Dong, Matthew Dixon, Panashe Kanengoni, Najeeb Kazmi, Pete Luferenko, Zhongjian Lv, Shikhar Sharma, Jonathan A. Weyn, and Siqi Xiang. Ms-nowcasting: Operational precipitation nowcasting with convolutional lstms at microsoft weather. *ArXiv*, abs/2111.09954, 2021. URL https://api.semanticscholar.org/CorpusID:244463010.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445, 2024.

Jie Liu, Lei Xu, and Nengcheng Chen. A spatiotemporal deep learning model st-lstm-sa for hourly rainfall forecasting using radar echo images. *Journal of Hydrology*, 2022. URL https://api.semanticscholar.org/CorpusID:247602986.

Chuyao Luo, Xinyue Zhao, Yuxi Sun, Xutao Li, and Yunming Ye. Predrann: The spatiotemporal attention convolution recurrent neural network for precipitation nowcasting. *Knowl. Based Syst.*, 239:107900, 2021. URL https://api.semanticscholar.org/CorpusID:245591327.

Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024.

Irina Malkin Ondík, Lukáš Ivica, Peter Šišan, Ivan Martynovskyi, David Šaur, and Ladislav Gaál. A concept of nowcasting of convective precipitation using an x-band radar for the territory of the zlín region (czech republic). In *Computer Science On-line Conference*, pages 499–514. Springer, 2022.

JS Marshall, Walter Hitschfeld, and KLS Gunn. Advances in radar weather. In *Advances in geophysics*, volume 2, pages 1–56. Elsevier, 1955.

Marta Martinkova and Jan Kysely. Overview of observed clausius-clapeyron scaling of extreme precipitation in midlatitudes. *Atmosphere*, 11(8):786, 2020.

Cristian Meo, Louis Mahon, Anirudh Goyal, and Justin Dauwels. αTC-VAE: On the relationship between disentanglement and diversity. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=ptXo0epLQo.

Cristian Meo, Ankush Roy, Mircea Lică, Junzhe Yin, Zeineb Bou Che, Yanbo Wang, Ruben Imhoff, Remko Uijlenhoet, and Justin Dauwels. Extreme Precipitation Nowcasting using Transformer-based Generative Models, March 2024b. URL http://arxiv.org/abs/2403.03929. arXiv:2403.03929 [cs].

Cristian Meo, Mircea Lica, Zarif Ikram, Akihiro Nakano, Vedant Shah, Aniket Rajiv Didolkar, Dianbo Liu, Anirudh Goyal, and Justin Dauwels. Masked generative priors improve world models sequence modelling capabilities, 2025. URL https://arxiv.org/abs/2410.07836.

National Weather Service. Vil density, n.d. URL https://www.weather.gov/lmk/vil_density. Accessed: 2025-05-07.

Aart Overeem and Ruben Imhoff. Archived 5-min rainfall accumulations from a radar dataset for the netherlands, 2020. URL https://data.4tu.nl/articles/_/12675278/1.

Aart Overeem, Iwan Holleman, and Adri Buishand. Derivation of a 10-year radar-based climatology of rainfall. *Journal of Applied Meteorology and Climatology*, 48(7):1448–1463, 2009.

C. Pierce, A. Seed, S. Ballard, D. Simonin, and Z. Li. Nowcasting. In *Doppler Radar Observations: Weather Radar, Wind Profiler, Ionospheric Radar, and Other Advanced Applications*. InTech, 2012. doi: 10.5772/39054.

Rachel Prudden, Samantha Adams, Dmitry Kangin, Niall Robinson, Suman Ravuri, Shakir Mohamed, and Alberto Arribas. A review of radar-based nowcasting of precipitation and applicable machine learning techniques. *arXiv preprint arXiv:2005.04988*, 2020.

Seppo Pulkkinen, Daniele Nerini, Andrés A Pérez Hortal, Carlos Velasco-Forero, Alan Seed, Urs Germann, and Loris Foresti. Pysteps: An open-source python library for probabilistic precipitation nowcasting (v1. 0). *Geoscientific Model Development*, 12(10):4185–4219, 2019.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

Shuhuai Ren, Shuming Ma, Xu Sun, and Furu Wei. Next block prediction: Video generation via semi-auto-regressive modeling. *arXiv preprint arXiv:2502.07737*, 2025.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Timothy J Schmit, Paul Griffith, Mathew M Gunshor, Jaime M Daniels, Steven J Goodman, and William J Lebair. A closer look at the abi on the goes-r series. *Bulletin of the American Meteorological Society*, 98(4):681–698, 2017.

Alan W Seed, Clive E Pierce, and Katie Norman. Formulation and evaluation of a scale decomposition-based stochastic precipitation nowcast scheme. *Water Resources Research*, 49 (10):6624–6641, 2013.

AW Seed. A dynamic and spatial scaling approach to advection forecasting. *Journal of Applied Meteorology and Climatology*, 42(3):381–388, 2003.

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. *Advances in neural information processing systems*, 28, 2015.

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. *Advances in neural information processing systems*, 30, 2017.

Zbynek Sokol, Jan Mejsnar, Lukas Pop, and Vojtech Bliznak. Probabilistic precipitation nowcasting based on an extrapolation of radar reflectivity and an ensemble approach. *Atmospheric Research*, 194:245–257, 2017. ISSN 0169-8095. doi: 10.1016/j.atmosres.2017.05.003.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL `https://arxiv.org/abs/1706.03762`. Version Number: 7.

Marc Veillette, Siddharth Samsi, and Christopher J. Mattioli. Sevir : A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in Neural Information Processing Systems*, 33:22009–22019, 2020. URL `https://api.semanticscholar.org/CorpusID:227222587`.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Junzhe Yin, Cristian Meo, Ankush Roy, Zeineh Bou Cher, Mircea Lică, Yanbo Wang, Ruben Imhoff, Remko Uijlenhoet, and Justin Dauwels. Precipitation nowcasting using physics informed discriminator generative models. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pages 967–971. IEEE, 2024. doi: 10.23919/EUSIPCO63174.2024.10715141.

Demin Yu, Xutao Li, Yunming Ye, Baoquan Zhang, Chuyao Luo, Kuai Dai, Rui Wang, and Xunlai Chen. DiffCast: A Unified Framework via Residual Diffusion for Precipitation Nowcasting, March 2024. URL `http://arxiv.org/abs/2312.06734`. arXiv:2312.06734 [cs].

Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

A Dataset Analysis

A.1 KNMI Dataset

The KNMI dataset contains radar-based precipitation estimates collected by two weather radars located in the Netherlands Overeem and Imhoff [2020]. The data has a spatial resolution of 1 km^2 and a temporal resolution of 5 minutes, covering the entire land area of the Netherlands. It spans the period from 2008 to 2018, during which the radar infrastructure underwent a significant upgrade.

The raw measurements recorded by these radars are in the form of *radar reflectivity*, which quantifies the amount of transmitted microwave energy reflected back after encountering precipitation particles. Reflectivity values are converted to precipitation rates using a standard Z–R relationship, given by:

$$Z_h = 200R^{1.6},$$

where Z_h is the horizontal radar reflectivity factor and R is the precipitation rate in mm/hr Marshall et al. [1955].

The dataset is highly imbalanced towards low/no precipitation events. To address this, we retain only those events with average precipitation above the 50th percentile for all subsequent experiments. This subset includes data spanning from 2008 to 2018 and provides a more informative, balanced foundation for model training and evaluation.

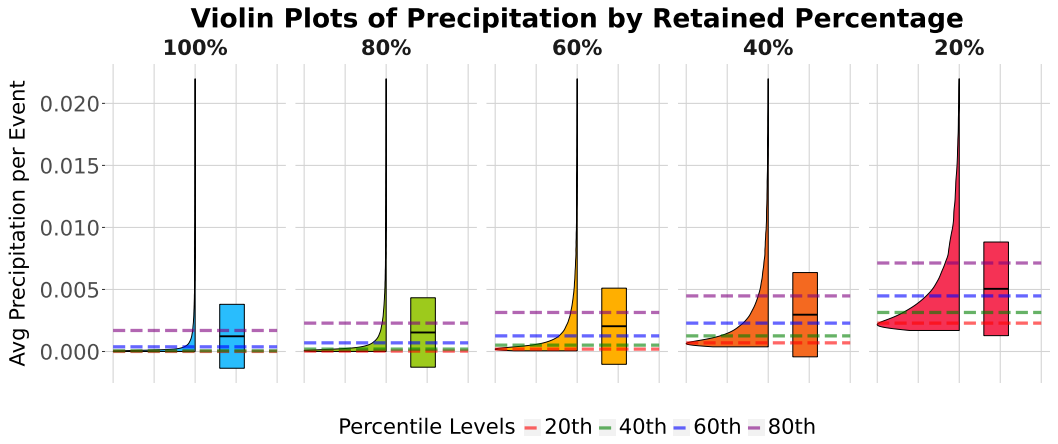

Figure 2: Violin plots of event average precipitation in the KNMI dataset.

A.2 SEVIR Dataset

The SEVIR dataset, introduced in Veillette et al. [2020], is a machine-learning-ready resource that aggregates multiple remote sensing modalities, including satellite and radar data. It consists of 4-hour weather events covering 384 km $\times$ 384 km regions across the continental United States, sampled every 5 minutes. The spatial resolution is 1 km^2 for most modalities, including Vertically Integrated Liquid (VIL).

SEVIR provides five sensing modalities: three channels from the GOES-16 (Geostationary Operational Environmental Satellite) system Schmit et al. [2017], VIL measurements, and data from the Geostationary Lightning Mapper (GLM) Goodman et al. [2013]. VIL, which is derived from radar reflectivity, estimates the total liquid water content in a vertical column of the atmosphere and is commonly used to assess intense precipitation events such as thunderstorms and hail National Weather Service [n.d.]. For this study, we focus on the VIL modality.

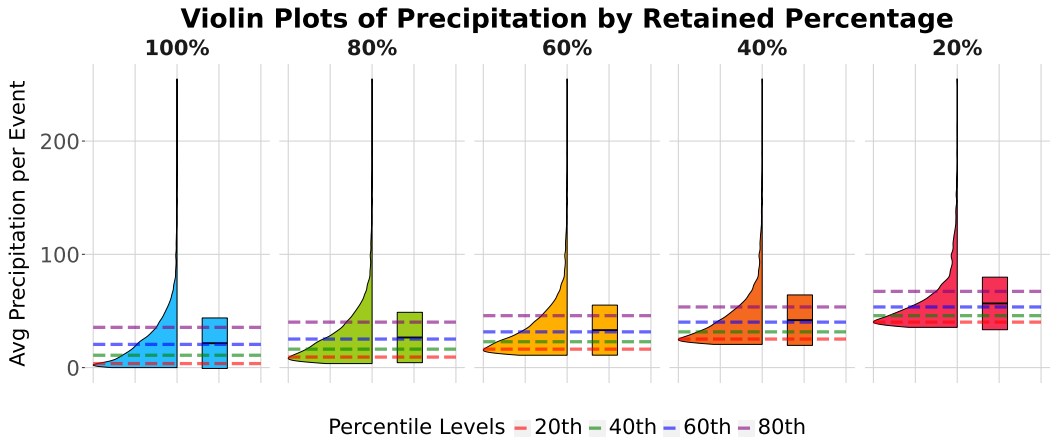

Figure 3: Violin plots of event average precipitation in the SEVIR dataset.

B Extended Results

B.1 Qualitative Case Studies

We compare two representative events per dataset, each showing two input frames ($t = -60$ and $t = 0$ min) and four forecast frames ($t = +30, +60, +120, +180$ min) from BlockGPT (ours), NowcastingGPT (token-level autoregression), and DiffCast+Phydnet (diffusion-based), against the ground truth.

KNMI. In Figure 4, BlockGPT captures the elongated rainband and its left-to-right advection, preserving embedded convective cores and their growth. While peak intensity is slightly overestimated at longer horizons, the morphology and displacement remain accurate. In contrast, NowcastingGPT and DiffCast+Phydnet miss the band structure and misplace intense cells; the diffusion pipeline also exhibits blobby artefacts and fails to recover the linear organisation. A similar pattern holds for the more challenging convective case in Figure 5, where rapid growth and relocation of high-intensity cells occur: BlockGPT reconstructs the evolving structure and localisation, whereas baselines struggle with both evolution and displacement.

SEVIR. For the linear convective system in Figure 6, BlockGPT reproduces structure and propagation more faithfully than both baselines. At far lead times, it tends to overestimate peak intensity, consistent with the higher errors seen in the continuous metrics. In the circular/rotational case of Figure 7, BlockGPT again tracks geometry and location best; baselines lose the organised shape. Occasional overestimation of the high-intensity core at longer horizons aligns with our quantitative findings on SEVIR.

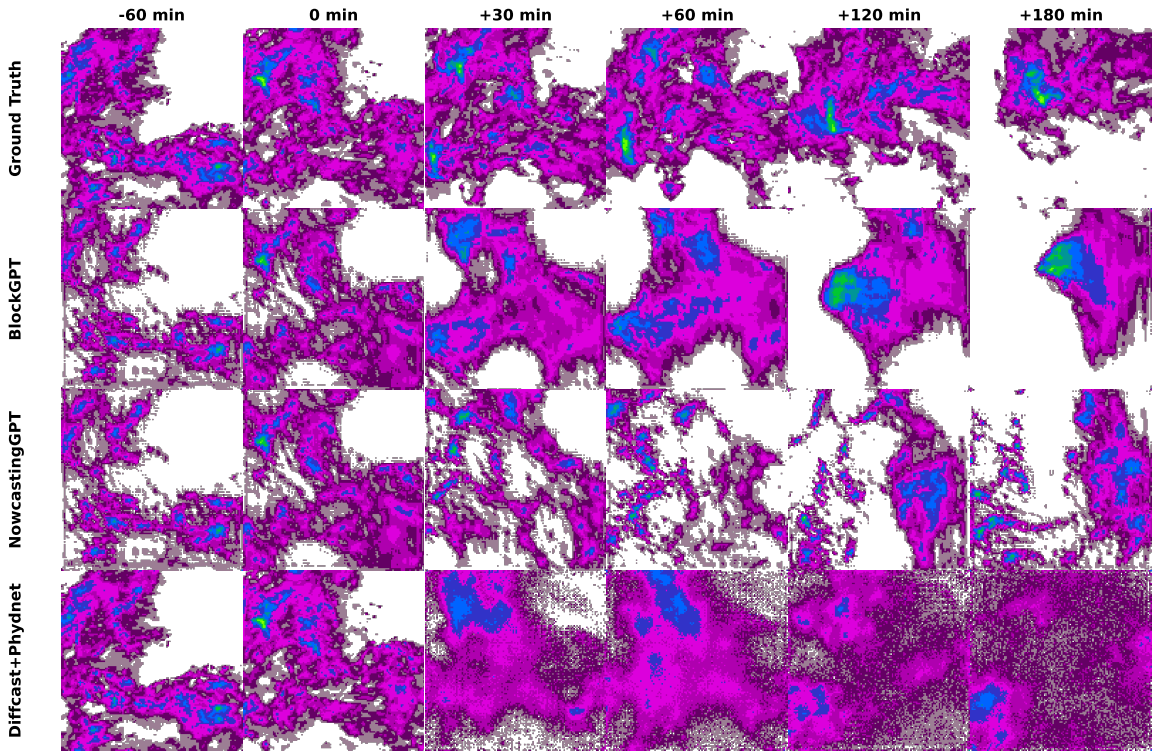

Figure 4: KNMI Event 1. Two input frames (-60, 0 min) and four forecasts ($+30$, $+60$, $+120$, $+180$ min). BlockGPT preserves the rainband morphology and advection but modestly overestimates the core intensity at long lead times; baselines miss the shape and location.

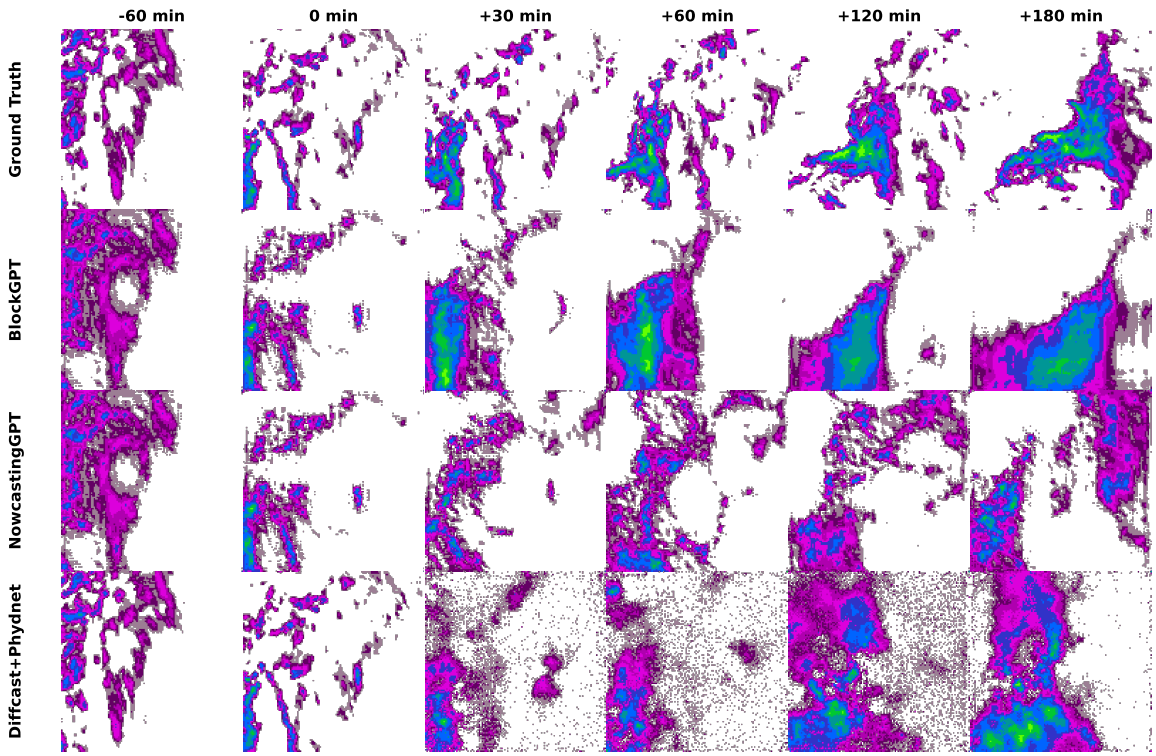

Figure 5: KNMI Event 2. BlockGPT follows the rapid structural changes and localisation of intense cells across lead times; baselines underperform, particularly for growth and displacement.

B.2 Continuous Score Metrics across Percentile Levels

We evaluate continuous scores for aggregated data and across intensity-conditioned percentile bins (0–20, 20–40, 40–60, 60–80, 80–95th).

KNMI. At low-percentile bins, BlockGPT shows relatively higher MSE/MAE than the baselines; errors drop markedly toward higher-percentile bins where accurate prediction is operationally most critical. Across all bins, PCC is consistently higher for BlockGPT. Aggregated over all intensities, BlockGPT attains lower MSE/MAE and higher PCC than both benchmarks, indicating overall superiority. We also observe larger uncertainty across seeds for BlockGPT, attributable to the batched tokeniser design which can amplify seed-to-seed variability. Aggregated behaviour is summarised in Figure 12.

SEVIR. Across bins, BlockGPT yields higher MSE/MAE than the baselines due to its strong sensitivity to high-intensity cores, which can be overestimated at longer lead times—incurring double-penalty effects from both intensity and displacement errors. Nevertheless, PCC remains competitive and typically exceeds DiffCast+Phydnet, indicating better spatial pattern fidelity despite larger amplitude errors. Aggregated trends are shown in Figure 13.

B.3 Catchment Analysis (KNMI only)

We assess event-detection skill over hydrologically critical KNMI subregions Imhoff et al. [2020] using ROC and AUC-ROC at thresholds 1, 2, and 8 $\mathrm{mm\,h^{-1}}$ for lead times 30, 60, 90, 120, 150, and 180 min. ROC curves are computed for each threshold; AUC summarises skill across false-positive rates. As shown in Figure 8, BlockGPT consistently outperforms both NowcastingGPT and DiffCast+Phydnet in AUC-ROC at all thresholds and lead times, evidencing robustness across intensity and horizon. As expected, detection skill diminishes with increasing lead time for all models, yet BlockGPT maintains the best performance across subregions. The ROC curves at a 1, 2 and 8 $\mathrm{mm}h^{-1}$ thresholds, stratified by lead time, are provided in Figure 9, Figure 10 and Figure 11.

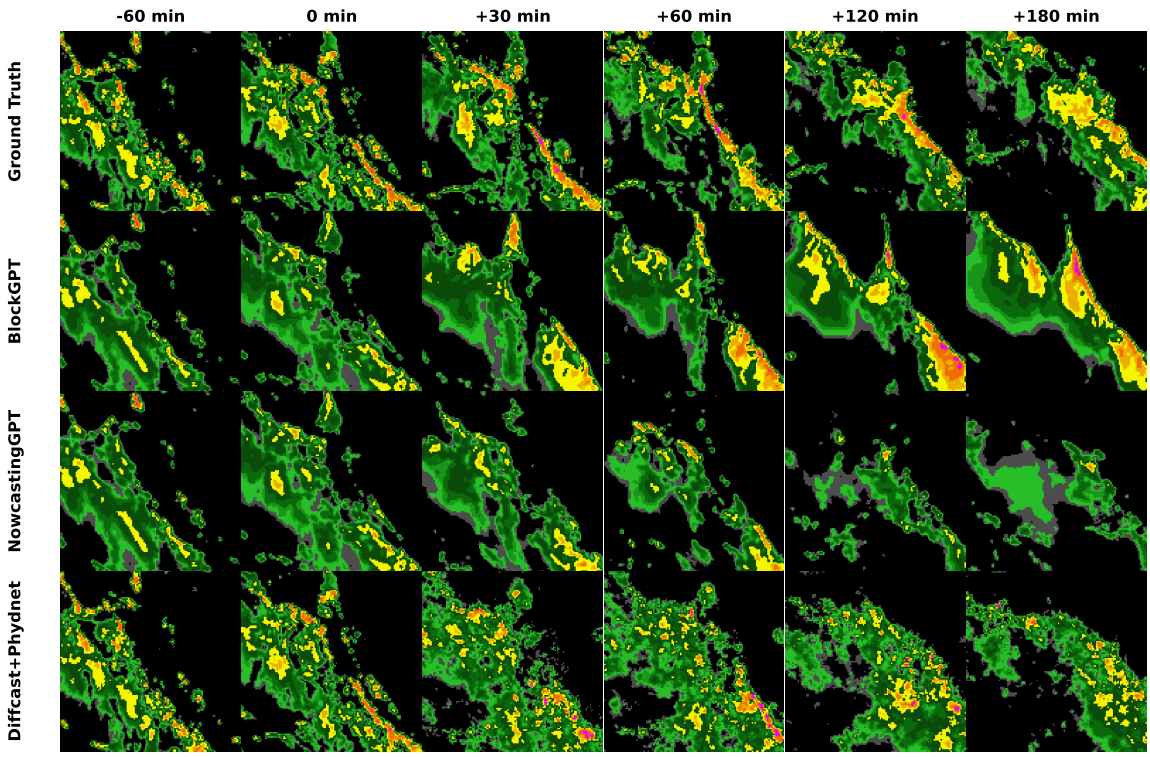

Figure 6: SEVIR Event 1 (linear system). BlockGPT best maintains structure and motion; a tendency to overestimate peak intensity emerges at longer lead times.

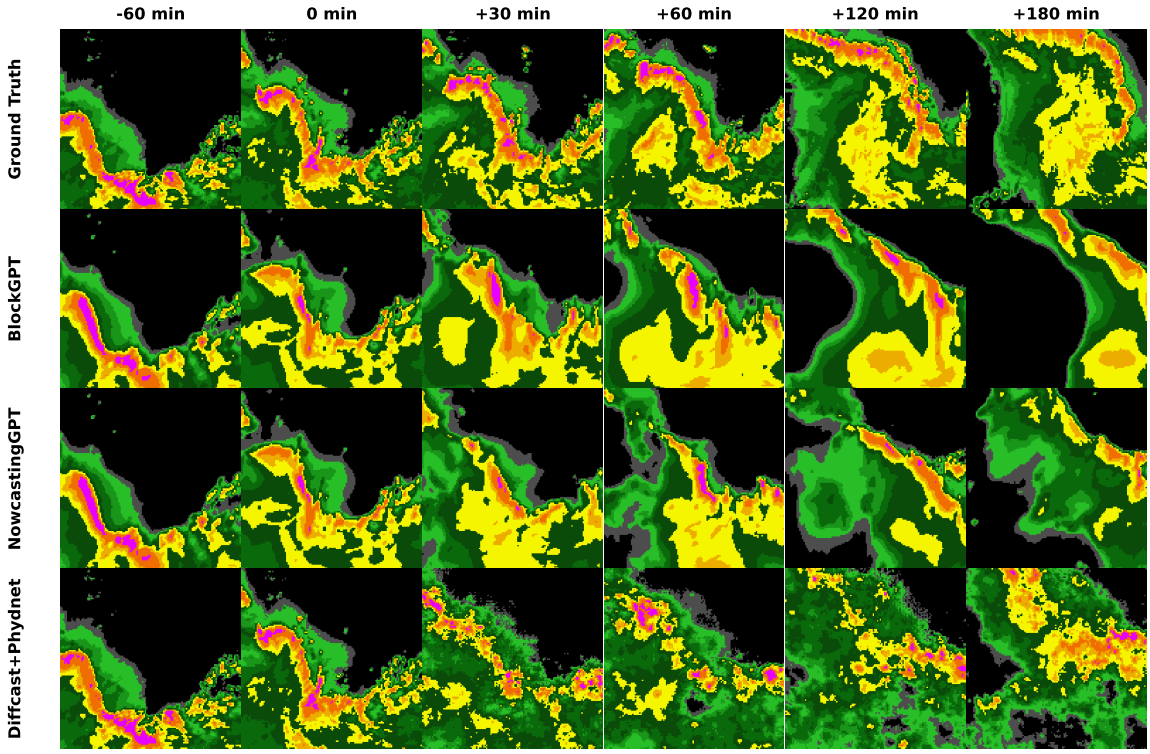

Figure 7: SEVIR Event 2 (circular organisation). BlockGPT best preserves the circular structure and its evolution; baselines lose shape and localisation.

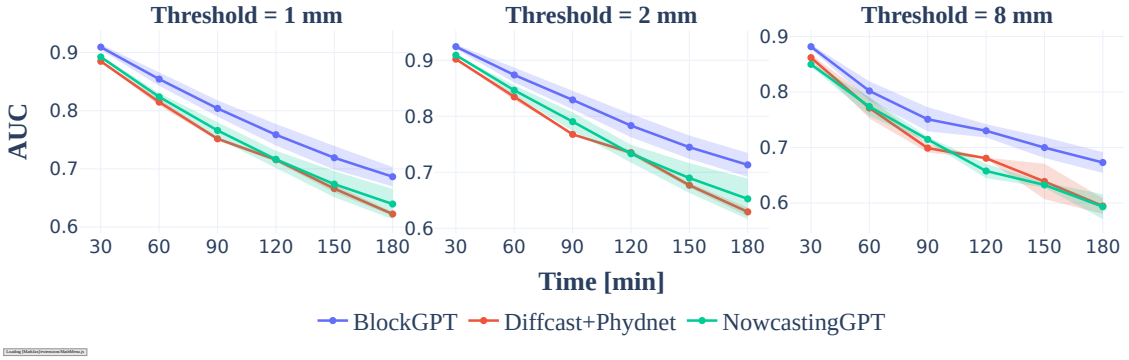

Figure 8: KNMI catchments: AUC-ROC over lead time for thresholds 1, 2, and 8 mm h^{-1}. Skill declines with lead time for all methods, but BlockGPT dominates across thresholds and horizons.

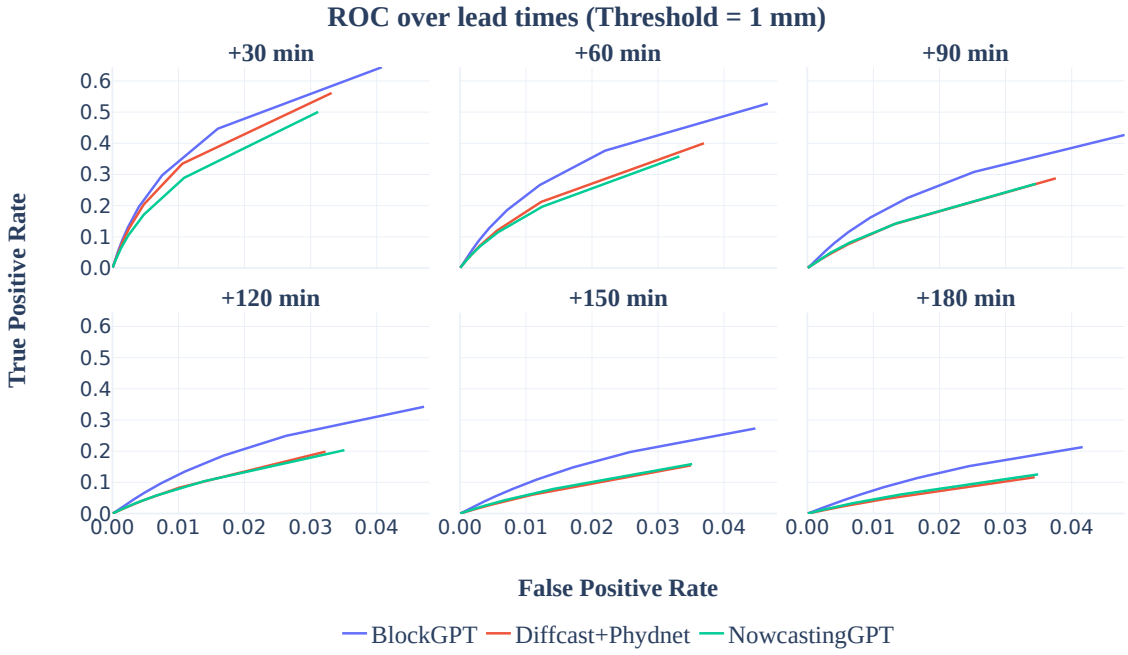

Figure 9: ROC curves across lead times for a 1 mm h^{-1} threshold on the KNMI (30 min) dataset. Panels correspond to +30, +60, +90, +120, +150, and +180 min lead times. Curves are averaged across seeds; performance improves as curves approach the top-left corner. Models compared: BlockGPT, DiffCast+Phydnet, and NowcastingGPT.

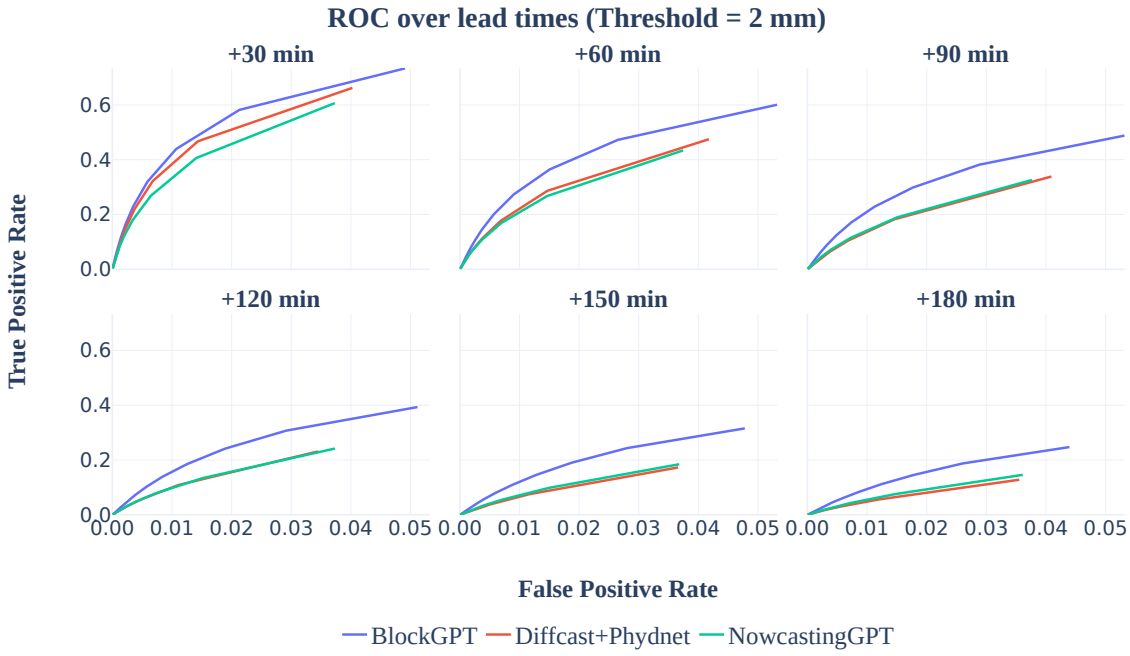

Figure 10: ROC curves across lead times for a 2 mm h^{-1} threshold on the KNMI (30 min) dataset. Panels correspond to +30, +60, +90, +120, +150, and +180 min lead times. Curves are averaged across seeds; performance improves as curves approach the top-left corner. Models compared: BlockGPT, DiffCast+Phydnet, and NowcastingGPT.

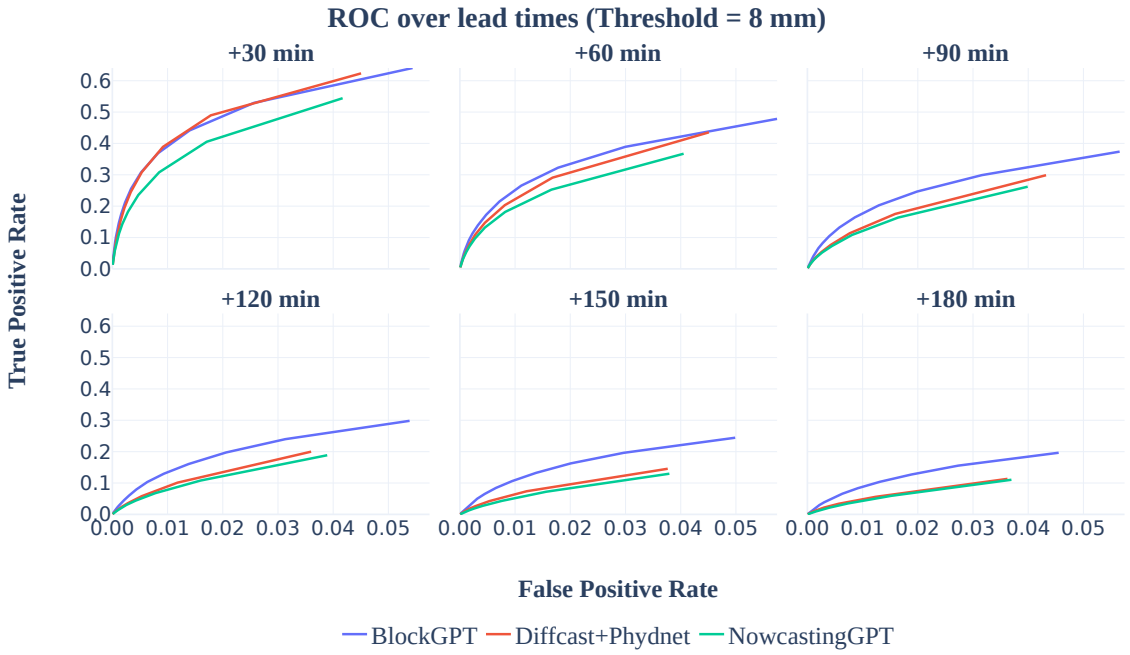

Figure 11: ROC curves across lead times for a 8 mm h^{-1} threshold on the KNMI (30 min) dataset. Panels correspond to +30, +60, +90, +120, +150, and +180 min lead times. Curves are averaged across seeds; performance improves as curves approach the top-left corner. Models compared: BlockGPT, DiffCast+Phydnet, and NowcastingGPT.

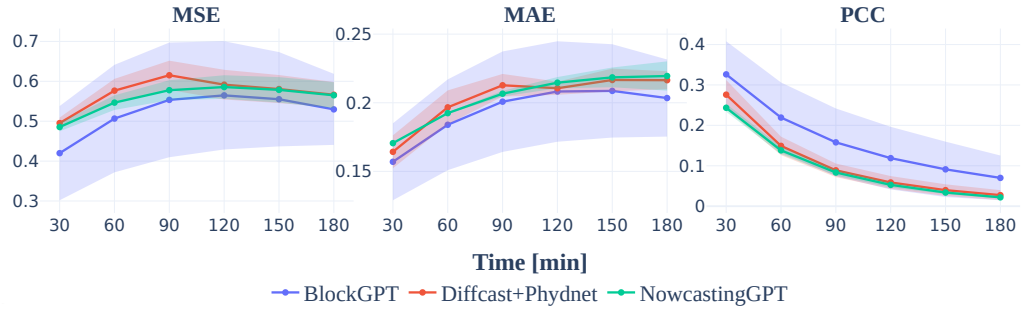

Figure 12: KNMI: Continuous metrics (MSE, MAE, PCC) by percentile bin (0–20, 20–40, 40–60, 60–80, 80–95th). BlockGPT exhibits superior PCC across bins; MSE/MAE reduce notably at higher bins. Aggregated scores favour BlockGPT overall.

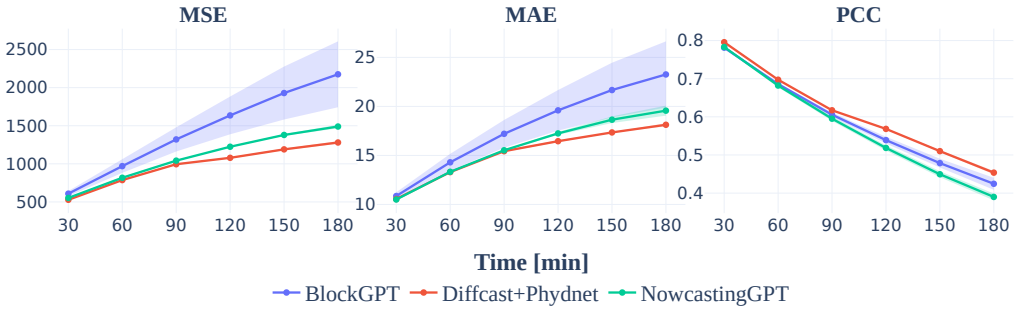

Figure 13: SEVIR: Continuous metrics (MSE, MAE, PCC) by percentile bin (0–20, 20–40, 40–60, 60–80, 80–95th). BlockGPT shows elevated MSE/MAE due to overestimation of high-intensity cores at long lead times, while maintaining favourable PCC relative to DiffCast+Phydnet.

C Model Architecture Details

C.1 VQ-GAN Training Details

The first stage of the BlockGPT pipeline compresses high-dimensional precipitation maps into a compact, discrete latent representation. For this, we employ a Vector Quantized-Generative Adversarial Network (VQ-GAN) Esser et al. [2021], which leverages an encoder-decoder framework with a discrete codebook $\mathcal{Z}$, and a discriminator $\mathcal{D}$, that discriminates between reconstructed and ground truth images. The encoder maps an input precipitation map x into a lower-resolution grid of feature vectors, preserving essential spatial information by reducing spatial dimensions while increasing feature channels. Specifically, the encoder consists of 5 downsampling layers, each containing 2 ResNet blocks, which progressively reduce the spatial resolution from 128×128 down to 8×8. The final stage of the encoder includes an attention block to better capture global feature relationships before quantization.

Each feature vector $\hat{z} = E(x)$ is then mapped to its closest entry in the learned codebook $\mathcal{Z}$ via an element-wise quantization step $q(\cdot)$:

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left(\arg\min_{z_k \in \mathcal{Z}} \|\hat{z} - z_k\| \right).$$

This process yields a grid of discrete latent tokens z_q for each input frame. The decoder, which mirrors the encoder's architecture, then reconstructs the precipitation map $\hat{x} = \text{Dec}(z_q)$ from these quantized tokens. The VQ-GAN is trained by optimizing a combination of reconstruction, commitment, and perceptual losses to ensure both high-fidelity reconstruction and a well-structured latent

space:

$$\mathcal{L}_{\text{VQ-VAE}} = \|x - \hat{x}\|_2^2 + \beta \|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 + \|\text{sg}[z_{\mathbf{q}}] - E(x)\|_2^2 + \mathcal{L}_{\text{perceptual}}(x, \hat{x}), \tag{3}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator, and the commitment loss (the third term) is weighted by a hyperparameter β. To ensure the generation of realistic maps, an adversarial loss from the discriminator $\mathcal{D}$ is added. Therefore, the GAN Loss ($\mathcal{L}_{\text{GAN}}$) loss is given by:

$$\mathcal{L}_{\text{GAN}} = \mathcal{L}_{\text{VQ-VAE}} + \lambda \mathbb{E}_{x \sim p(x)}[\log \mathcal{D}(x) + \log(1 - \mathcal{D}(\hat{x}))] \tag{4}$$

where the term λ is an adaptive weight calculated from the gradients of the perceptual and GAN losses to balance their contributions during training.

D Extended Related Work

Precipitation nowcasting, a sub-field of spatio-temporal forecasting, presents unique challenges due to the chaotic and stochastic evolution of weather systems. While traditional methods based on physical principles, like the advection-based PySTEPS Pulkkinen et al. [2019], are well-established, they often struggle to model complex, non-linear dynamics Ravuri et al. [2021]. In contrast, deep learning (DL) models have demonstrated a remarkable ability to learn these patterns directly from vast amounts of radar data Shi et al. [2015], Ravuri et al. [2021]. The shift of paradigm that led DL models to succeed was casting precipitation nowcasting as a video prediction problem Bi et al. [2023], Bai et al. [2022], Luo et al. [2021], where given an input spatio-temporal sequence of N frames $\boldsymbol{x}_{\text{in}} \in \mathbb{R}^{N \times H \times W \times C}$, H, W denote the spatial resolution and C represents the image channels or the different type of measurements (e.g., radar, heat maps, etc), the goal is to predict the next M frames $\boldsymbol{x}_{\text{out}} \in \mathbb{R}^{M \times H \times W \times C}$. In this section we present the literature related to precipitation nowcasting models, the main related field of this paper.

D.1 Video Prediction Architectures

A prominent architectural pattern in modern generative video modeling involves a three-stage process: (1) a compression stage that encodes high-dimensional frames into a discrete latent space, (2) a generation stage that models the dynamics of these latent representations, and (3) a diffusion step that models the residuals that were not captured by the video prediction backbone.

The compression is typically handled by a Vector Quantized-Variational Autoencoder (VQ-VAE) Van Den Oord et al. [2017], which learns a codebook of visual tokens. The generation is then performed by a powerful sequence model, often an autoregressive Transformer Vaswani et al. [2017], which learns to predict the next token in a sequence. This approach was popularized for general video generation by VideoGPT Yan et al. [2021] and adapted for precipitation nowcasting by NowcastingGPT Meo et al. [2024b]. These models typically flatten the 2D grid of spatial tokens into a 1D sequence and predict them one-by-one.

However, this token-level autoregression imposes a flawed inductive bias by assuming a causal, sequential relationship between tokens that are spatially correlated. This creates an ill-posed learning problem that can result in spatially fragmented outputs and suffers from slow inference speeds due to its sequential nature Tian et al. [2024], Luo et al. [2024].

To address these limitations, recent works have shifted towards generating tokens in larger chunks or in parallel. Some methods have explored non-autoregressive generation using masking strategies Chang et al. [2022]. Our work is most closely related to the emerging paradigm of block-level autoregression, where an entire block of tokens—or in our case, an entire frame—is predicted at each time step Li et al. [2024], Yu et al. [2023]. This approach, explored in models like Next Block Prediction Ren et al. [2025], allows for bidirectional self-attention within a frame to capture spatial structures effectively, while maintaining a causal autoregressive structure across time to model temporal evolution. By adopting this frame-level prediction strategy, BlockGPT aims to overcome the efficiency and coherence issues of prior token-based nowcasting models.

E Experimental Setup Details

E.1 Training Configuration

All models are trained with Adam (learning rate 1×10^{-4}) using a 10,000-step warmup. Training runs for 500,000 steps with batch size 8. Evaluation is performed on a held-out test set unused during training/validation.

E.2 Evaluation Metrics

To assess the quality of predicted precipitation sequences, we employ continuous (value-based) and categorical (event-based) metrics. Let an event be a sequence of T frames

$$\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{T_c}, \mathbf{X}_{T_c+1}, \ldots, \mathbf{X}_T\} = \{\mathcal{X}_{\text{context}}, \mathcal{X}_{\text{target}}\},$$

where $\mathbf{X}_t \in \mathbb{R}^{H \times W}$ is the radar field at time t. Given predictions $\hat{\mathcal{X}}_{\text{target}} = \{\hat{\mathbf{X}}_{T_c+1}, \ldots, \hat{\mathbf{X}}_T\}$ and targets $\mathcal{X}_{\text{target}} = \{\mathbf{X}_{T_c+1}, \ldots, \mathbf{X}_T\}$, we define:

Continuous Metrics

These quantify Jolliffe and Stephenson [2003] amplitude accuracy over the full spatiotemporal domain.

Mean Squared Error (MSE). Penalizes large deviations quadratically; sensitive to outliers and thus captures severe intensity errors.

$$\text{MSE} = \frac{1}{(T - T_c)HW} \sum_{t=T_c+1}^{T} \sum_{i=1}^{H} \sum_{j=1}^{W} \big(\hat{\mathbf{X}}_t[i,j] - \mathbf{X}_t[i,j]\big)^2. \tag{5}$$

Mean Absolute Error (MAE). Measures median-like deviation; robust to outliers and interpretable in physical units.

$$\text{MAE} = \frac{1}{(T - T_c)HW} \sum_{t=T_c+1}^{T} \sum_{i=1}^{H} \sum_{j=1}^{W} \big|\hat{\mathbf{X}}_t[i,j] - \mathbf{X}_t[i,j]\big|. \tag{6}$$

Pearson Correlation Coefficient (PCC). Assesses linear association and phase coherence independent of bias and scale.

$$\text{PCC} = \frac{\sum_{t,i,j} \big(\hat{\mathbf{X}}_t[i,j] - \bar{\hat{\mathbf{X}}}\big)\big(\mathbf{X}_t[i,j] - \bar{\mathbf{X}}\big)}{\sqrt{\sum_{t,i,j} \big(\hat{\mathbf{X}}_t[i,j] - \bar{\hat{\mathbf{X}}}\big)^2} \sqrt{\sum_{t,i,j} \big(\mathbf{X}_t[i,j] - \bar{\mathbf{X}}\big)^2}}, \tag{7}$$

where $\bar{\hat{\mathbf{X}}}$ and $\bar{\mathbf{X}}$ are means over all target pixels and times.

Categorical Metrics

These evaluate Jolliffe and Stephenson [2003]event detection (e.g., exceedance of a threshold τ). We binarize frames as

$$\hat{\mathbf{Y}}_t[i,j] = \mathbb{1}\big[\hat{\mathbf{X}}_t[i,j] \geq \tau\big], \tag{8}$$

$$\mathbf{Y}_t[i,j] = \mathbb{1}\big[\mathbf{X}_t[i,j] \geq \tau\big]. \tag{9}$$

Over all (t,i,j) in the target period, the contingency counts are

$$\text{TP} = \sum_{t,i,j} \mathbb{1}\big[\hat{\mathbf{Y}}_t[i,j] = 1,\ \mathbf{Y}_t[i,j] = 1\big], \tag{10}$$

$$\text{FP} = \sum_{t,i,j} \mathbb{1}\big[\hat{\mathbf{Y}}_t[i,j] = 1,\ \mathbf{Y}_t[i,j] = 0\big], \tag{11}$$

$$\text{FN} = \sum_{t,i,j} \mathbb{1}\big[\hat{\mathbf{Y}}_t[i,j] = 0,\ \mathbf{Y}_t[i,j] = 1\big], \tag{12}$$

$$\text{TN} = \sum_{t,i,j} \mathbb{1}\big[\hat{\mathbf{Y}}_t[i,j] = 0,\ \mathbf{Y}_t[i,j] = 0\big]. \tag{13}$$

	Observed 1 (event)	Observed 0 (no event)
Forecast 1 (event)	TP (hit)	FP (false alarm)
Forecast 0 (no event)	FN (miss)	TN (correct rejection)

Table 1: Contingency table for threshold-exceedance events.

Critical Success Index (CSI). Fraction of correctly predicted events among all observed or forecast events; penalizes misses and false alarms.

$$\mathrm{CSI} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{14}$$

False Alarm Ratio (FAR). Proportion of forecast events that did not occur; lower is better.

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}}. \tag{15}$$

Receiver Operating Characteristic (ROC) and AUC. The ROC curve Jolliffe and Stephenson [2003], Pulkkinen et al. [2019] measures discrimination skill for threshold-exceedance events by varying a decision threshold γ on forecast probabilities or continuous scores derived from the predicted frames and comparing to the observed binary targets in (9). For each γ, compute the hit rate (probability of detection, POD) and the false-alarm rate (probability of false detection, POFD/FPR) from the contingency counts in (10)–(13):

$$\mathrm{POD}(\gamma) = \frac{\mathrm{TP}(\gamma)}{\mathrm{TP}(\gamma) + \mathrm{FN}(\gamma)}, \tag{16}$$

$$\mathrm{POFD}(\gamma) = \frac{\mathrm{FP}(\gamma)}{\mathrm{FP}(\gamma) + \mathrm{TN}(\gamma)}. \tag{17}$$

Plotting $\mathrm{POD}(\gamma)$ against $\mathrm{POFD}(\gamma)$ yields the ROC curve; better discrimination pushes the curve toward the upper-left corner $(\mathrm{POFD} = 0, \mathrm{POD} = 1)$, indicating that predicted exceedances align with observed exceedances while rarely triggering on non-events (i.e., high hits, few false alarms). The area under the curve summarizes potential skill,

$$\mathrm{AUC} = \int_0^1 \mathrm{POD}\big(\mathrm{POFD}\big)\, d\,\mathrm{POFD}, \tag{18}$$

with $\mathrm{AUC} = 0.5$ for no-skill and $\mathrm{AUC} = 1$ for perfect discrimination; note that POFD in (17) is not the same as FAR.

E.3 Statistical Significance

All scores are aggregated over 3 random seeds; where shown, shaded bands denote ± 1 standard deviation to convey sampling uncertainty.

F Model Parameters and Training Time

In this section, we summarize the training configurations and compute profiles of the three models compared in this work. A key goal in our design of BlockGPT was to match or outperform the benchmark models in terms of training time, while scaling model capacity up to the point of overfitting. The table below presents a comparison of parameter counts and training durations across all models.

Our model, BlockGPT, was designed under the constraint of maintaining a training budget that is no greater than that of our benchmarks. Within this constraint, we maximize model capacity by increasing the number of parameters up to the point of overfitting or until the training time matches that of the benchmarks. This approach ensures a fair and efficient comparison while allowing us to explore the benefits of larger model capacity within realistic computational limits. We retain the original model configurations of DiffCast+Phydnet and NowcastingGPT. For the latter, we retain the same model parameters as those in the checkpoints in the github repository of Meo et al. [2024b].

Model	Parameters	Training Time	Hardware / Epochs
DiffCast+Phydnet	49.35M	$\sim$15 hours	2 $\times$ A100 GPUs / 20 epochs
NowcastingGPT	150M	$\sim$6 hours	2 $\times$ A100 GPUs / 20 epochs
BlockGPT (Ours)	103.37M	$\sim$6 hours	2 $\times$ A100 GPUs / 20 epochs

Table 2: Training time and parameter comparison across all models.

The embedding dimension however, was originally only 128. We therefore retrain with the same embedding dimension as ours, for fair comparison.

BlockGPT's model configuration is as follows:

Table 3: Essential architecture specifications for VQGAN and BlockGPT Transformer.

Component	Parameter	Value
VQGAN	Codebook Size	1024
	Latent Channels	128
	Attention Resolution	[8]
	Dropout	0.2
	Tokens per Frame	64
BlockGPT (Transformer)	Number of Layers	8
	Number of Heads	8
	Embedding Size	1024
	Token Block Size	576
	Vocabulary Size	1024

G Implementation Details

G.1 Code Availability

The implementation of BlockGPT and all experimental code will be made publicly available upon publication. The codebase includes training scripts, evaluation metrics, and pre-trained models for reproducibility.

G.2 Computational Requirements

Training BlockGPT requires approximately 8 GPUs with 32GB memory each for 500,000 steps. Inference can be performed on a single GPU, making it suitable for real-time applications.

A key advantage of BlockGPT is its computational efficiency. As shown in Table 4, BlockGPT is significantly faster than both benchmarks. On the 30-minute task, it is 27× faster than NowcastingGPT and 31× faster than DiffCast+Phydnet. On the 5-minute task, it is 31× faster than NowcastingGPT and 10× faster than DiffCast+Phydnet. These results indicate that frame-level autoregression not only improves performance but also greatly enhances computational efficiency.

Model	Inference Time (s)
NowcastingGPT	7.09
DiffCast+Phydnet	8.17
BlockGPT	0.26

Table 4: Inference time (in seconds) per batch for each model.

G.3 Hyperparameter Tuning

We conducted extensive hyperparameter tuning for all models to ensure fair comparison. The final hyperparameters were selected based on validation performance on a held-out validation set.