PRECIPITATION NOWCASTING OF SATELLITE DATA USING PHYSICALLY- ALIGNED NEURAL NETWORKS

Antônio Catão, Melvin Poveda, Leonardo Voltarelli & Paulo Orenstein
Instituto Nacional de Matemática Pura e Aplicada
Rio de Janeiro, RJ, Brazil
{antonio.catao,melvin.poveda,leonardo.voltarelli,pauloo}@impa.br

ABSTRACT

Accurate short-term precipitation forecasts predominantly rely on dense weather-radar networks, limiting operational value in places most exposed to climate extremes. We present TUPANN, a satellite-only model that decomposes the forecast into physically meaningful components: a variational encoder-decoder infers motion and intensity fields from recent imagery under optical-flow supervision, a lead-time-conditioned MaxViT evolves the latent state, and a differentiable advection operator reconstructs future frames. We evaluate TUPANN on both GOES-16 and IMERG data, in up to four distinct climates at 10–180-min lead times using the CSI metric. Comparisons against optical-flow, deep learning and hybrid baselines show that TUPANN achieves the best or second-best skill in most settings, with pronounced gains at higher thresholds. The model produces smooth, interpretable motion fields aligned with numerical optical flow and runs in near real time. These results indicate that physically aligned learning can provide nowcasts that are skillful, transferable and global.

1 INTRODUCTION

Extreme precipitation events are projected to become more frequent and intense under climate change (K. et al., 2023). Nowcasting—forecasting on time horizons up to 6 h at high spatial resolution—is critical for early warnings and disaster management. While numerical weather prediction has improved steadily, its finite resolution and latency limit the accuracy of short-term precipitation forecasts. Recent advances in machine learning have shown that deep networks can outperform traditional numerical models in precipitation nowcasting when trained on high-resolution radar data. However, reliance on radar restricts their applicability to radar-rich regions, leaving large parts of South America, Africa and Asia underserved. Furthermore, purely data-driven architectures often struggle with physical interpretability: they may produce realistic-looking precipitation maps while neglecting physically consistent motion fields, hindering forecasters' trust and operational uptake.

This paper addresses both accessibility and interpretability by leveraging geostationary satellites, which provide global coverage in near real-time, and by incorporating explicit physical structure into the neural network. We present TUPANN (Transferable and Universal Physics-Aligned Nowcasting Network), a model that uses only satellite-derived precipitation fields and decomposes the forecasting problem into physically motivated submodules. TUPANN comprises a variational encoder–decoder trained under optical-flow supervision to recover motion and intensity fields, a lead-time-conditioned transformer to evolve latent states, and a differentiable advection operator to reconstruct future frames. A strong physical alignment is done by explicitly penalizing the encoder-decoder output to match results from optical flows algorithms, which numerically infers motion fields. We evaluate TUPANN on data from four climate regimes and report critical success indices (CSI) across lead times from 10 to 180 minutes and various precipitation thresholds.

2 DATA AND STUDY REGIONS

The primary data source used is the GOES-R Advanced Baseline Imager Rain Rate Quantitative Precipitation Estimation (RRQPE) (GOES-R Algorithm Working Group and GOES-R Program Of-

fice, 2018). RRQPE provides precipitation estimates over the Americas every 10 min at 2 km spatial resolution with a latency of approximately 5 min, enabling real-time nowcasting. This product is highly correlated with rain-related bands and has been validated against ground radars and the GPM CORRA dataset (Agrawal et al., 2025), highlighting the value of predicting geostationary satellite observations. We use RRQPE from January 2020 to December 2023 and build a dataset of rain events as defined in detail in the Appendix A. To test generalization across data sources we also use the Integrated Multi-satellitE Retrievals for GPM (IMERG) Final Run product (Huffman et al., 2014) (see Appendix A). IMERG provides precipitation estimates every 30 min at 10 km resolution and is widely used in remote sensing research. To evaluate model performance across different climates we select four 512 km $\times$ 512 km subregions of the GOES-16 domain centered on Rio de Janeiro (Brazil), La Paz (Bolivia), Manaus (Brazil) and Miami (USA). These regions span coastal, high-altitude, rainforest and subtropical environments.

3 METHOD

TUPANN forecasts precipitation fields from a sequence of past satellite images $X_{-T:0} \in \mathbb{R}^{(T+1)\times n\times n}$, where n denotes the spatial resolution (i.e., number of pixels per dimension) and $T+1$ is the number of past observations. It produces predicted fields $\widehat{X}_{1:T_f} \in \mathbb{R}^{T_f\times n\times n}$, where T_f is the forecast horizon. The training procedure is sequential: the VED module is initially trained to infer the first set of motion and intensity fields, then its weights are fixed and used for the training of the MaxVit module. A diagram of the architecture is provided in Figure 1.

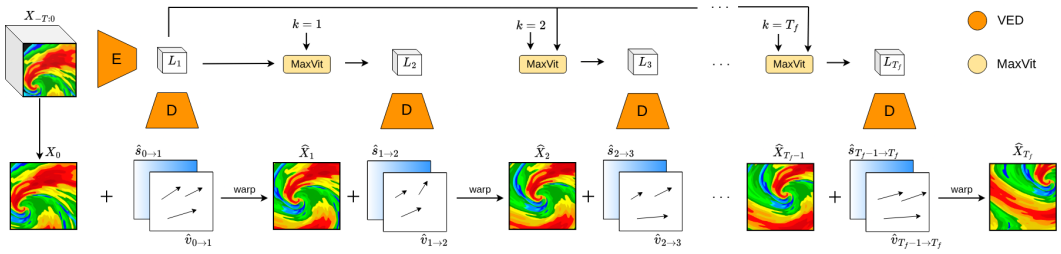

Figure 1: TUPANN architecture. The VED and MaxViT modules displayed are learned; motion fields and the final predictions are extrapolated through a warp function.

Even though MaxViT offers linear complexity in the image size, the choice of such encoder-decoder architecture is guided by the idea that, apart from compressing the images, the VED is also responsible for learning the dynamics of a single step evolution. MaxViT, on the other hand, learns to extrapolate the dynamics to further lead times.

3.1 VARIATIONAL ENCODER–DECODER

We use a variational encoder–decoder to learn an efficient representation of the precipitation evolution. Instead of reconstructing the input images as in classical variational autoencoders, our VED outputs motion fields $\widehat{v}_{0\to1} \in \mathbb{R}^{2\times n\times n}$ and intensity corrections $\widehat{s}_{0\to1} \in \mathbb{R}^{n\times n}$ given the past sequence $X_{-T:0}$. To enforce physically plausible motion we compute the ground truth motion fields $v_{0\to1}$ applying an optical flow algorithm to $X_{-\tilde{T}+2:1}$, where $\tilde{T}$ is the context length provided to the optical flow algorithm. After that, the ground truth intensity correction $s_{0\to1}$ is obtained by subtracting the advected frame $\tilde{X}_1$, obtained using $v_{0\to1}$, from the true frame X_1 (see Figure 2). Thus, the estimated fields $\widehat{v}_{0\to1}$ and $\widehat{s}_{0\to1}$ can be used to extrapolate the last observed frame X_0 to an estimate $\widehat{X}_1$ of the next frame via an advection operator.

To supervise the predicted $\widehat{v}_{0\to1}$, ℓ_1 and cosine-similarity losses are used with respect to $v_{0\to1}$. The intensity discrepancies between $s_{0\to1}$ and $\widehat{s}_{0\to1}$ are penalized via ℓ_1 loss. Finally, a Kullback-Leibler divergence term is added to ensure the regularity of the learned latent space. The VED loss is

$$\begin{aligned}\text{Loss}_{\text{VED}}(\widehat{s}_{0\to1}, \widehat{v}_{0\to1}, s_{0\to1}, v_{0\to1}) := \lambda_{\text{int}}\, \ell_1(s_{0\to1}, \widehat{s}_{0\to1}) + \lambda_{\text{motion}}\, \ell_1(v_{0\to1}, \widehat{v}_{0\to1}) \\ + \lambda_{\cos}\, \text{CosSimilarity}(v_{0\to1}, \widehat{v}_{0\to1}) + \lambda_{\text{KL}}\, KL(p, \widehat{p}_\theta).\end{aligned} \tag{1}$$

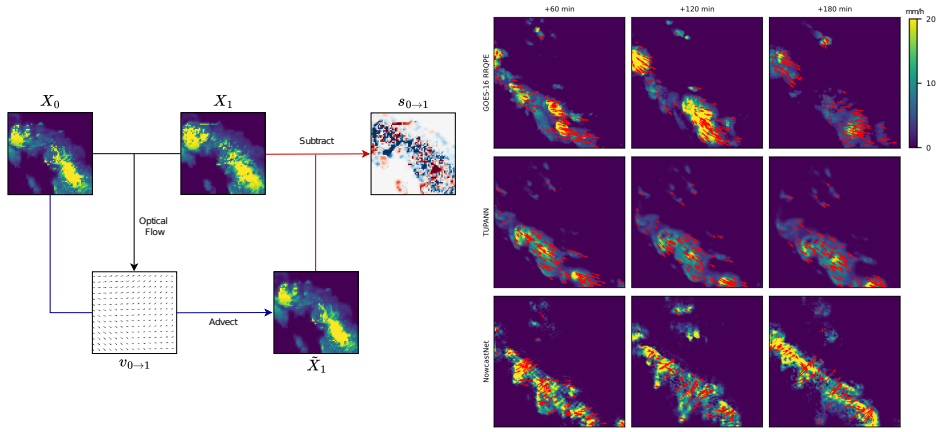

Figure 2: Left: An optical flow method yields ground truth motion fields ($v_{0\to1}$) through a pair of past (X_0) and future (X_1) images. X_0 is advected to obtain $\tilde{X}_1$. Ground truth intensity ($s_{0\to1}$) is obtained by subtracting $\tilde{X}_1$ from X_1. Right: Comparison of motion fields. Top row shows ground-truth DARTS motion fields. TUPANN yields fields that are smoother than other benchmarks, such as NowcastNet, and that align with physical intuition.

Here, $\widehat{p}_\theta$ is the latent distribution inferred by the encoder, p is a standard normal distribution and $\lambda_{\text{int}}, \lambda_{\text{motion}}, \lambda_{\cos}, \lambda_{\text{KL}}$ are hyperparameters tuned on the validation set. This loss encourages accurate motion fields, intensity corrections and latent regularity. At the suggestion of an anonymous reviewer, we have also experimented directly computing the loss on the extrapolated image $\widehat{X}_1$, leading to better CSI performance at the cost of interpretability. See Section C.2 for details.

An optical flow algorithm is able to infer motion fields between two images, and is thus essential to obtain the derived ground-truth motion and intensity fields (e.g., $v_{0\to1}, s_{0\to1}$) used in equation 1. We consider two options: Lucas–Kanade (LK), which solves a local least-squares problem under the assumption of small displacements, and DARTS, a spectral method tailored to radar imagery that solves the optical-flow equation in Fourier space. For TUPANN, we have taken the choice of optical flow method as a hyperparameter; below we use DARTS for GOES-16 and LK for IMERG results.

3.2 MaxViT

Given the latent representation L_1 from the VED, a visual transformer evolves the latent state forward in time. We adopt MaxViT (Tu et al., 2022), which combines local and grid attention to efficiently capture global context while avoiding quadratic attention cost.

To predict the latent state at lead time k, we condition the transformer on k via one-hot encoding and linear embedding, yielding $L_k = \text{VT}(L_1, k)$, $k = 2, \ldots, T_f$, where $\text{VT}(\cdot)$ represents the MaxViT model. Unlike recurrent decoding, this conditioning enables the same transformer to produce all lead times while reducing memory overhead (Andrychowicz et al., 2023). Applying the VED decoder to L_k yields motion and intensity fields $\left(\widehat{v}_{k-1\to k}, \widehat{s}_{k-1\to k}\right) = D(L_k)$, where $D(\cdot)$ is the decoder module of the VED. Thus, all necessary elements to predict the sequence recursively are obtained.

Following NowcastNet (Zhang et al., 2023), we implement a fixed differentiable advection operator that can reconstruct future precipitation frames using the predicted motion and intensity fields. Thus, given a frame $\widehat{X}_{k-1}$, motion field $\widehat{v}_{k-1\to k}$ and intensity field $\widehat{s}_{k-1\to k}$, the extrapolated frame $\widehat{X}_k$ is

$$\widehat{X}_k = \text{warp}\left(\widehat{v}_{k-1\to k}, \widehat{s}_{k-1\to k}, \widehat{X}_{k-1}\right). \tag{2}$$

We compute the target loss on the original image. We assume that $\widehat{X}_{k-1} = X_{k-1}$ in equation (2) to avoid a costly recursive loss and calculate the ℓ_1 loss between $\widehat{X}_k$ and X_k. Thus,

$$\text{Loss}_{\text{MaxViT}} = \ell_1\left(\text{warp}\left(\widehat{v}_{k-1\to k}, \widehat{s}_{k-1\to k}, X_{k-1}\right), X_k\right). \tag{3}$$

The gradients of this loss will flow through the VED decoder module and the MaxViT transformer. The VED is pre-trained separately, thus optimizing this loss only affects the MaxViT modules.

4 EXPERIMENTS AND RESULTS

TUPANN is compared with four baselines: PySTEPS (LK) and PySTEPS (DARTS) (Pulkkinen et al., 2019); Earthformer (Gao et al., 2022); NowcastNet (Zhang et al., 2023) and CasCast (Gong et al., 2024). For further details on the baselines and training, see Appendix A.

Table 1 presents CSI scores (Hogan & Mason, 2011) across the four study regions. TUPANN consistently ranks first or second for each metric. In Rio de Janeiro, it achieves the highest CSI at all thresholds; NowcastNet is the closest competitor, followed by Earthformer. In Miami, TUPANN again dominates most metrics. In Manaus and La Paz, Earthformer and NowcastNet obtain slightly better CSI for low thresholds, but TUPANN leads for higher thresholds and is therefore better at forecasting extreme rainfall. The 64 mm/h CSI values are small across models, reflecting the rarity of such intense events, yet TUPANN's scores remain the highest. Overall, the table highlights TUPANN's performance across different climate regimes and precipitation thresholds. Beyond excellent metrics, we also visually validate TUPANN's interpretability and physical alignment through the motion fields it generates, as shown in Figure 2. Further evaluations are present in Appendix B.

Table 1: Aggregated CSI metrics for GOES-16 data across cities. **Bold** values denote the best, underlined values the second best. TUPANN obtains state-of-the-art performance at most thresholds and regions, particularly for high rain-rate events.

Model	CSI-M ↑		CSI_4 ↑		CSI_8 ↑		CSI_{16} ↑		CSI_{32} ↑		CSI_{64} ↑	
	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4
	Rio de Janeiro											
Earthformer	0.237	0.222	0.326	0.320	0.287	0.236	0.326	0.312	0.238	0.237	0.009	0.006
NowcastNet	0.244	0.269	0.313	0.374	0.282	**0.293**	0.318	0.325	0.247	0.278	0.059	0.074
PySTEPS (LK)	0.165	0.169	0.242	0.262	0.212	0.195	0.226	0.226	0.142	0.156	0.005	0.008
PySTEPS (DARTS)	0.166	0.166	0.231	0.243	0.216	0.191	0.229	0.228	0.140	0.152	0.013	0.015
CasCast	0.170	0.187	0.308	0.343	0.205	0.249	0.164	0.162	0.159	0.156	0.016	0.027
TUPANN (ours)	**0.259**	**0.277**	**0.330**	**0.384**	**0.289**	0.289	**0.330**	**0.336**	**0.274**	**0.287**	**0.072**	**0.090**
	Miami											
Earthformer	0.141	0.126	**0.274**	0.270	0.180	0.160	0.154	0.122	0.097	0.078	0.000	0.000
NowcastNet	0.137	0.160	0.248	0.299	0.170	0.207	0.128	0.139	0.097	0.106	0.040	0.047
PySTEPS (LK)	0.113	0.116	0.188	0.202	0.133	0.136	0.120	0.111	0.079	0.079	0.045	0.053
PySTEPS (DARTS)	0.112	0.113	0.189	0.202	0.135	0.138	0.118	0.107	0.073	0.071	0.044	0.046
CasCast	0.146	0.170	0.258	0.298	0.188	**0.229**	0.144	0.167	0.117	0.135	0.020	0.028
TUPANN (ours)	**0.169**	**0.187**	0.267	**0.312**	**0.189**	0.211	**0.177**	**0.177**	**0.135**	**0.141**	**0.079**	**0.094**
	Manaus											
Earthformer	0.276	0.256	**0.355**	0.341	**0.323**	0.297	**0.316**	0.292	0.265	0.245	0.124	0.104
NowcastNet	0.253	0.278	0.323	0.366	0.296	0.324	0.283	0.303	0.233	0.258	0.130	0.137
PySTEPS (LK)	0.200	0.196	0.258	0.266	0.237	0.233	0.218	0.212	0.160	0.156	0.125	0.112
PySTEPS (DARTS)	0.197	0.194	0.259	0.268	0.239	0.235	0.219	0.213	0.158	0.154	0.109	0.099
CasCast	0.265	0.286	0.344	**0.377**	0.303	**0.333**	0.295	0.307	0.260	0.269	0.126	0.141
TUPANN (ours)	**0.290**	**0.293**	0.339	0.367	0.316	0.321	0.315	**0.312**	**0.278**	**0.274**	**0.200**	**0.193**
	La Paz											
Earthformer	0.303	0.270	**0.337**	0.312	**0.329**	0.281	**0.359**	0.319	0.323	0.291	0.167	0.146
NowcastNet	0.291	0.301	0.330	**0.376**	0.303	0.321	0.321	0.315	0.300	0.296	0.202	0.197
PySTEPS (LK)	0.212	0.208	0.248	0.250	0.243	0.230	0.247	0.237	0.197	0.195	0.126	0.131
PySTEPS (DARTS)	0.225	0.218	0.263	0.262	0.262	0.244	0.264	0.250	0.206	0.201	0.127	0.132
CasCast	0.228	0.235	0.309	0.337	0.251	0.270	0.245	0.237	0.232	0.222	0.101	0.111
TUPANN (ours)	**0.314**	**0.317**	0.336	0.363	0.327	**0.323**	0.350	**0.340**	**0.327**	**0.322**	**0.232**	**0.239**

5 CONCLUSION

We have presented TUPANN, a physically aligned neural network for precipitation nowcasting using satellite imagery. TUPANN's modular design — combining a variational encoder–decoder supervised by optical flow, and a transformer capable of evolving the latent representation according to physical constraints — yields interpretable motion fields and competitive forecast skill. Extensive experiments on GOES-11 and IMERG data across four climates show that TUPANN matches or surpasses state-of-the-art baselines, particularly at high precipitation thresholds. With its low latency and reliance on globally available satellites, TUPANN supports equitable access to short-term rainfall forecasts and provides a foundation for operational applications in radar-sparse regions.

REFERENCES

Shreya Agrawal, Mohammed Alewi Hassen, Emmanuel Asiedu Brempong, Boris Babenko, Fred Zyda, Olivia Graham, Di Li, Samier Merchant, Santiago Hincapie Potes, Tyler Russell, Danny Cheresnick, Aditya Prakash Kakkirala, Stephan Rasp, Avinatan Hassidim, Yossi Matias, Nal Kalchbrenner, Pramod Gupta, Jason Hickey, and Aaron Bell. An operational deep learning system for satellite-based high-resolution global nowcasting, 2025. URL https://arxiv.org/abs/2510.13050.

Marcin Andrychowicz, Lasse Espeholt, Di Li, Samier Merchant, Alexander Merose, Fred Zyda, Shreya Agrawal, and Nal Kalchbrenner. Deep learning for day forecasts from sparse observations, 2023. URL https://arxiv.org/abs/2306.06079.

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. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

GOES-R Algorithm Working Group and GOES-R Program Office. Noaa goes-r series advanced baseline imager (abi) level 2 rainfall rate / qpe. https://doi.org/10.7289/V5W66J21, 2018. URL https://doi.org/10.7289/V5W66J21.

Junchao Gong, Lei Bai, Peng Ye, Wanghan Xu, Na Liu, Jianhua Dai, Xiaokang Yang, and Wanli Ouyang. Cascast: skillful high-resolution precipitation nowcasting via cascaded modelling. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Robin J. Hogan and Ian B. Mason. *Deterministic Forecasts of Binary Events*, chapter 3, pp. 31–59. John Wiley & Sons, Ltd, 2011. ISBN 9781119960003. doi: https://doi.org/10.1002/9781119960003.ch3. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119960003.ch3.

G. Huffman, D. Bolvin, D. Braithwaite, K. Hsu, R. Joyce, and P. Xie. Integrated multi-satellite retrievals for gpm (imerg), version 4.4, 2014. URL ftp://arthurhou.pps.eosdis.nasa.gov/gpmdata/.

Marvel K., W. Su, R. Delgado, S. Aarons, A. Chatterjee, M.E. Garcia, Z. Hausfather, Katharine Hayhoe, D.A. Hence, E.B. Jewett, A. Robel, D. Singh, A. Tripati, R.S. Vose, A. Khan, Allison Crimmins, Christopher Avery, D.R. Easterling, K.E. Kunkel, and T.K. Maycock. *Fifth National Climate Assessment: Chapter 2 Climate Trends*. 11 2023. doi: 10.7930/NCA5.2023.CH2.

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*, 12(10):4185–4219, 2019. doi: 10.5194/gmd-12-4185-2019. URL https://gmd.copernicus.org/articles/12/4185/2019/.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. MaxViT: Multi-axis Vision Transformer. In *Computer Vision – ECCV 2022*, pp. 459–479. Springer, Cham, Switzerland, November 2022. ISBN 978-3-031-20053-3. doi: 10.1007/978-3-031-20053-3_27.

Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I. Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 619 (7970):526–532, 2023. doi: 10.1038/s41586-023-06184-4. URL https://doi.org/10.1038/s41586-023-06184-4.

APPENDIX

Note: while the submission template instructions asked for a separate pdf file for the appendix, the submission platform only allowed for a single file, so we include the appendix below.

A Implementation details

A.1 Baseline models and tuning

In our experiments, TUPANN is compared with four baselines from different nowcasting paradigms:

- PySTEPS (LK) and PySTEPS (DARTS) (Pulkkinen et al., 2019): optical-flow baselines that estimate a motion field (LK: local Lucas–Kanade; DARTS: DFT-based spectral) from recent frames and then semi-Lagrangianly advect the precipitation field forward. As purely physical extrapolation methods, they are strong for very short lead times;
- Earthformer (Gao et al., 2022): a space-time Transformer for Earth-system data that uses Cuboid Attention (local block attention with global tokens) in a hierarchical encoder–decoder to predict future frames;
- NowcastNet (Zhang et al., 2023): a hybrid model combining a U-Net–based learnable semi-Lagrangian advection (Evolution Network) with a physics-conditioned generative network trained with a temporal discriminator to inject high-resolution convective structure;
- CasCast (Gong et al., 2024): a cascaded scheme that first uses a deterministic predictor (e.g., Earthformer) to capture mesoscale evolution, then conditions a latent-space diffusion transformer on that coarse forecast to generate small-scale features and improve extreme-precipitation skill.

For TUPANN and Earthformer, we tune learning rate, dropout rate and loss weights on the validation set by maximizing mean CSI in the city of Rio de Janeiro, and use these for the other cities. Hyperparameters for NowcastNet and CasCast are mostly those presented in their original paper (see Section A.2). The optimizer for all models is Adam. After selecting the best values, we retrain on the combined training and validation data and evaluate on a held-out test set. Training uses a single NVIDIA A100 GPU, and inference typically takes under two seconds per forecast (for all 18 lead times).

A.2 Model Hyperparameters

The hyperparameters for Earthformer and TUPANN (both VED and MaxViT) are shown below. Hyperparameters Evolution Network/NowcastNet were the ones selected in the original paper (Zhang et al., 2023). Since CasCast uses a Denoising Transformer (DiT) model trained with a different image size (384x384), slight adaptations were made to support the 256x256 images used in this work. The hyperparameters modified were input_size to 32 and hidden_size to 512; other hyperparameters were chosen as in their original paper (Gong et al., 2024). All remaining hyperparameters and early stopping criteria were selected by maximizing the mean CSI value in the validation set during training.

Table 2: Model hyperparameters for VED, Earthformer, and MaxViT. Architectural and loss parameters are detailed in Section 3.

(a) VED

Parameter	Value
batch_size	8
learning_rate	0.0001
channels	128
embed_dim	4
reduc_factor	4
λ_{cos}	0.00165
λ_{KL}	1.0e-06
λ_{motion}	0.0033
λ_{int}	0.995
dropout	0.2

(b) Earthformer

Parameter	Value
batch_size	4
learning_rate	0.0001
num_global_vectors	6
num_heads	2
base_units	64

(c) MaxViT

Parameter	Value
batch_size	8
learning_rate	0.0001
MaxViT_depth	4
MaxViT_dim	64

A.3 RAIN EVENTS DATASET SELECTION

The datasets used to train and evaluate our models comprise a subsample of rainy windows drawn from either the GOES-16 RRQPE or the IMERG products. We define a rainy window as follows. For each 10-min timestamp t from 2020-01-01 00:00 UTC to 2023-12-31 23:50 UTC, we (i) form a symmetric 60-min window $[t - 30\,\text{min},\ t + 30\,\text{min}]$; (ii) compute the spatiotemporal precipitation accumulation over that window (10-min steps over all grid points); and (iii) if the accumulation exceeds a threshold τ, label the larger window $[t - 4\,\text{hours}, t + 4\,\text{hours}]$ as a rainy window. Finally, we merge rainy windows that intersect. The threshold $\tau = 120{,}000$ was chosen empirically to balance excluding near-dry periods against obtaining a dataset large enough for effective learning. Using a symmetric ± 4-hour window ensures that events include both onset and dissipation phases (from no precipitation to mild or heavy precipitation and back).

B FURTHER EXPERIMENTS

B.1 ADDITIONAL GOES-16 RESULTS

The graphs in Figure 3 show mean CSI (averaged across thresholds) versus lead time. TUPANN maintains the highest or second-highest skill across all lead times; the advantage over NowcastNet grows for early lead times, reflecting the benefit of explicit motion supervision and the efficiency of lead-time conditioning (see also Figure 4).

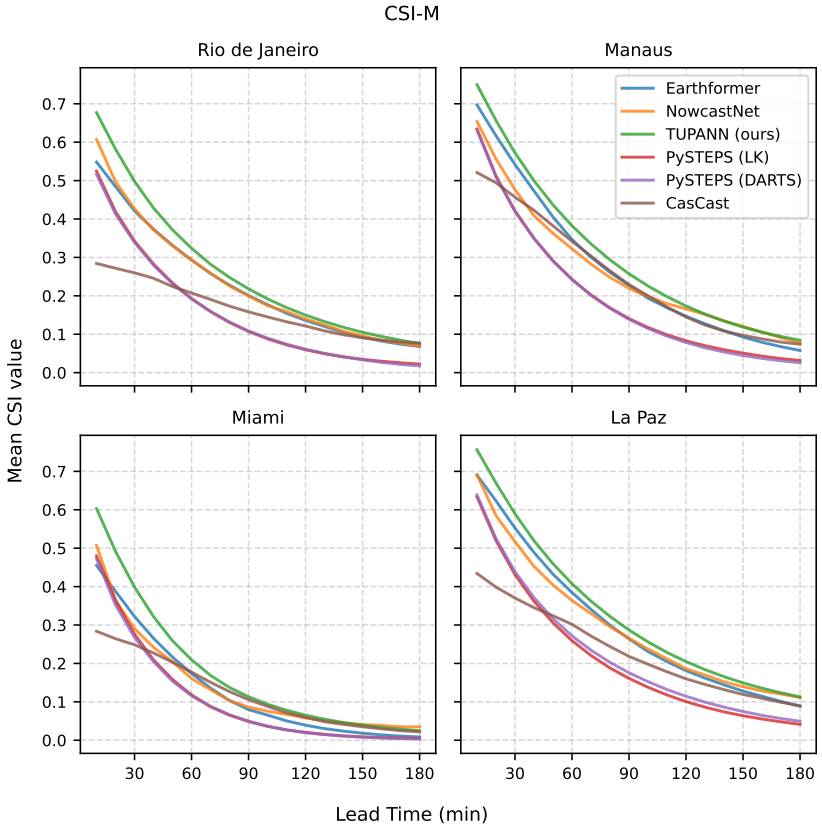

Figure 3: Mean CSI (CSI–M) versus lead time for the four study regions using GOES-16 data. TUPANN consistently outperforms baselines across lead times.

Beyond aggregated metrics, Figure 6 illustrates a TUPANN prediction for a rain event in Manaus, compared with NowcastNet, CasCast and Earthformer. Generative models such as NowcastNet and CasCast produce detailed textures but may introduce artifacts, whereas TUPANN and Earthformer

yield smoother predictions. Despite the blurred appearance, TUPANN captures the timing and location of heavy rain more accurately, leading to higher CSI values.

Aditionally, TUPANN's interpretability stems from its explicitly learned motion fields. Figure 4 compares motion fields predicted by TUPANN and NowcastNet (which relies on an Evolution Network submodule for its motion fields). The TUPANN fields are smooth and closely resemble the numerical optical flow computed by DARTS, whereas the baselines' fields exhibit unrealistic patterns. This underscores the benefit of supervising motion fields directly.

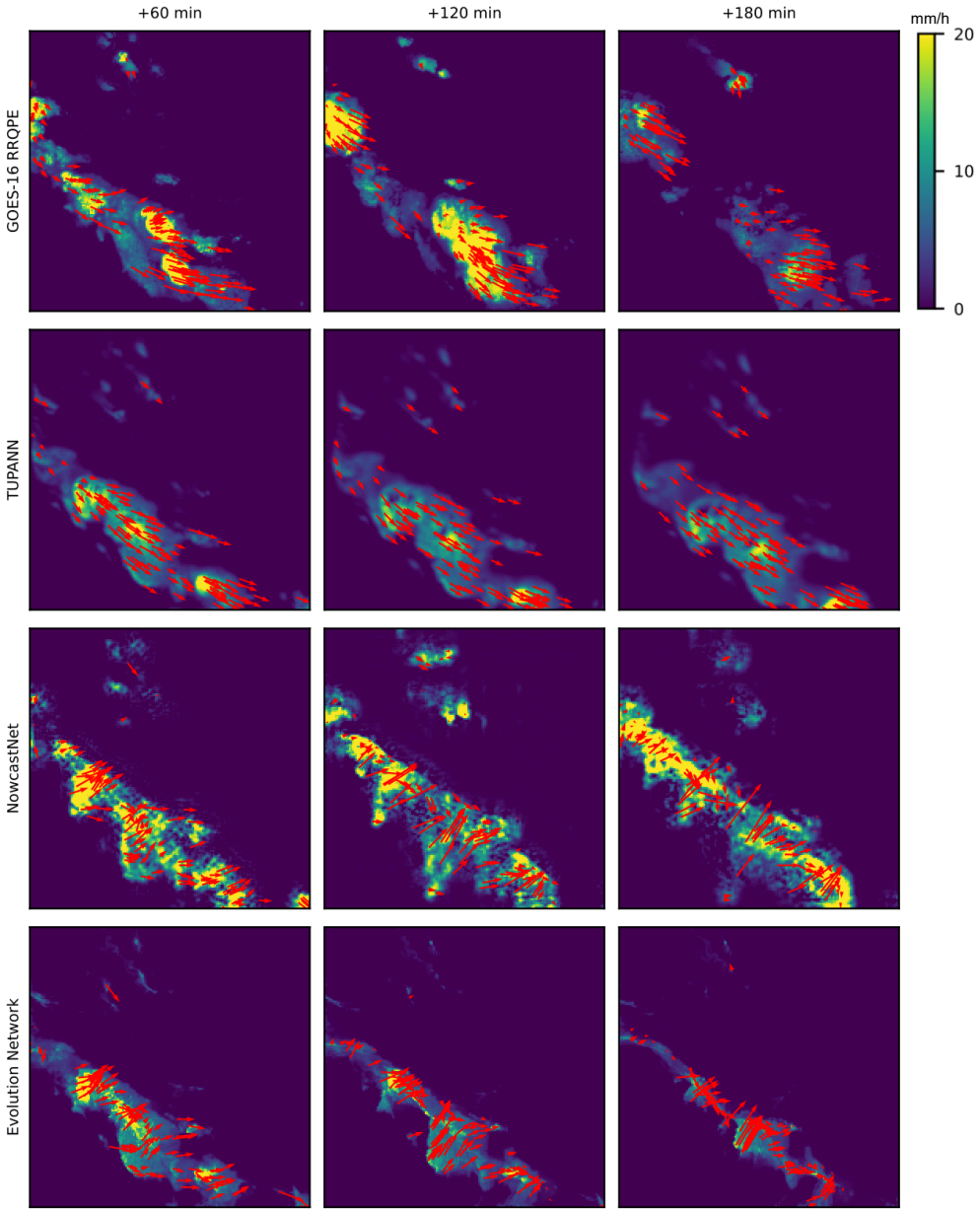

Figure 4: Comparison of motion fields. Top row: ground-truth DARTS motion fields for future frames. Second row: motion fields from TUPANN. Subsequent rows: motion fields estimated by NowcastNet/Evolution Network. TUPANN yields smoother fields that align with physical intuition.

Table 3 summarizes aggregated HSS metrics for the four cities using GOES-16 data. TUPANN achieves the best scores for high thresholds and remains competitive for lower thresholds, mirroring the CSI behaviour.

Table 3: Aggregated HSS metrics for GOES-16 data across cities. Bold values denote the best, underlined values the second best. TUPANN excels at high thresholds and is second or first at lower thresholds.

Model	HSS-M ↑	HSS_4 ↑	HSS_8 ↑	HSS_{16} ↑	HSS_{32} ↑	HSS_{64} ↑
	Rio de Janeiro					
Earthformer	0.360	<u>0.473</u>	<u>0.438</u>	<u>0.488</u>	0.382	0.017
NowcastNet	<u>0.373</u>	0.454	0.429	0.478	<u>0.394</u>	<u>0.112</u>
PySTEPS (LK)	0.266	0.369	0.341	0.365	0.247	0.010
PySTEPS (DARTS)	0.268	0.356	0.347	0.369	0.244	0.025
CasCast	0.266	0.438	0.321	0.271	0.270	0.032
TUPANN (ours)	**0.393**	**0.473**	**0.439**	**0.492**	**0.428**	**0.135**
	Miami					
Earthformer	0.230	**0.412**	0.299	<u>0.265</u>	0.176	0.000
NowcastNet	0.224	0.368	0.277	0.223	0.177	0.076
PySTEPS (LK)	0.195	0.299	0.227	0.213	0.147	<u>0.087</u>
PySTEPS (DARTS)	0.192	0.301	0.231	0.210	0.135	0.083
CasCast	<u>0.237</u>	0.388	<u>0.304</u>	0.247	<u>0.208</u>	0.039
TUPANN (ours)	**0.277**	<u>0.398</u>	**0.309**	**0.298**	**0.237**	**0.146**
	Manaus					
Earthformer	<u>0.417</u>	**0.502**	**0.476**	**0.472**	<u>0.414</u>	0.220
NowcastNet	0.384	0.457	0.437	0.429	0.372	<u>0.228</u>
PySTEPS (LK)	0.319	0.385	0.369	0.349	0.271	0.222
PySTEPS (DARTS)	0.314	0.386	0.371	0.350	0.269	0.195
CasCast	0.401	<u>0.486</u>	0.447	0.444	0.406	0.222
TUPANN (ours)	**0.435**	0.479	<u>0.464</u>	<u>0.469</u>	**0.430**	**0.333**
	La Paz					
Earthformer	<u>0.454</u>	**0.487**	**0.486**	**0.523**	<u>0.485</u>	0.285
NowcastNet	0.439	0.472	0.451	0.479	0.458	<u>0.335</u>
PySTEPS (LK)	0.340	0.378	0.381	0.390	0.326	0.223
PySTEPS (DARTS)	0.356	0.398	0.405	0.412	0.339	0.225
CasCast	0.351	0.441	0.381	0.381	0.370	0.183
TUPANN (ours)	**0.468**	<u>0.482</u>	<u>0.481</u>	<u>0.512</u>	**0.490**	**0.376**

B.2 IMERG RESULTS

Table 4 compares TUPANN to baselines on the IMERG dataset for Rio de Janeiro. We also include GAN-TUPANN, a variante of the proposed model which adds a GAN head to TUPANN outputs. Without pooling (POOL1), TUPANN achieves the best CSI across all thresholds. With pooling (POOL4), generative models (NowcastNet, GAN-TUPANN) slightly outperform TUPANN at low thresholds but TUPANN remains competitive and leads at higher thresholds. Figure 5 plots CSI–M versus lead time, showing TUPANN's superior performance at most lead times and small gaps only at 150 min. These results demonstrate that TUPANN generalizes to coarser spatial resolution and longer latency datasets, where its physics-aligned architecture can become slightly less beneficial.

B.3 VISUAL INSPECTION OF MODEL PREDICTIONS

Figure 6 includes predictions for several models at a given moment in time. While TUPANN and EarthFormer show relatively blurred predictions, note NowcastNet and CasCast add several artifacts to its predictions. Overall, TUPANN achieves a reasonable trade-off between good evaluation metrics and reasonable precipitation plots. Figure 7 also includes precipitation plots for the IMERG dataset, where the temporal and spatial resolution of the image is coarser.

Table 4: Aggregated CSI metrics for Rio de Janeiro using IMERG data. Bold denotes the best, underlined the second best. Without pooling TUPANN is clearly superior; with pooling, generative baselines perform slightly better at low thresholds, but TUPANN remains competitive overall.

Model	CSI-M ↑		CSI_4 ↑		CSI_8 ↑		CSI_{16} ↑		CSI_{32} ↑		CSI_{64} ↑	
	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4
Earthformer	0.153	0.141	0.361	0.343	0.270	0.243	0.130	0.114	0.006	0.006	0.000	0.000
NowcastNet	**0.209**	0.271	0.387	0.453	0.322	0.384	0.229	0.304	**0.103**	**0.201**	0.001	**0.013**
PySTEPS (LK)	0.114	0.099	0.300	0.269	0.191	0.164	0.075	0.061	0.002	0.003	0.000	0.000
PySTEPS (DARTS)	0.107	0.096	0.287	0.262	0.179	0.157	0.069	0.057	0.002	0.003	0.000	0.000
CasCast	0.180	0.229	0.362	0.420	0.288	0.339	0.188	0.248	0.061	0.139	0.000	0.000
TUPANN (ours)	**0.218**	0.248	**0.414**	0.454	**0.344**	0.379	**0.241**	0.280	0.087	0.123	**0.005**	0.006
GAN-TUPANN	0.210	**0.274**	0.391	**0.461**	0.327	**0.393**	0.234	**0.313**	0.095	0.199	0.002	0.004

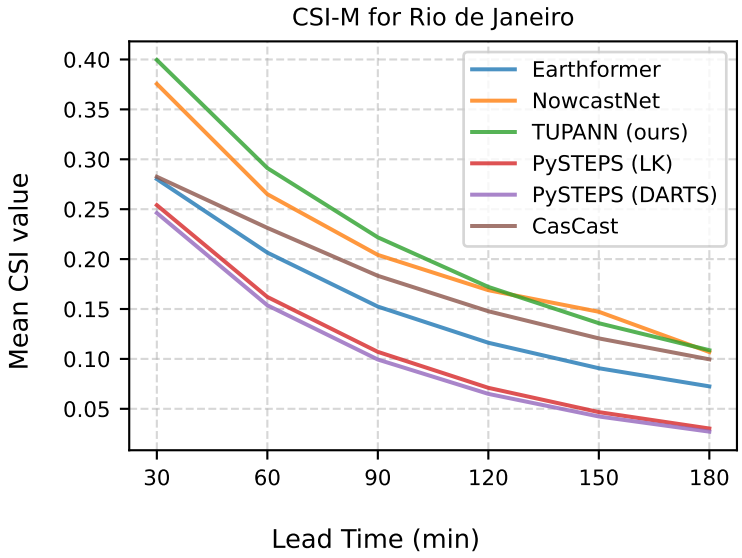

Figure 5: Mean CSI versus lead time for IMERG data in Rio de Janeiro. TUPANN outperforms baselines at most lead times; NowcastNet overtakes slightly at 150 min but lags at shorter lead times.

B.4 Spectral Analysis

Following Zhang et al. (2023), we perform a PSD analysis based on radially-averaged Fourier coefficients. In Figure 8, we provide these metrics (first row) along with an analog computed on the motion fields produced by TUPANN, NowcastNet (through its Evolution Network submodule), and PySTEPS (second row). This was done by performing a radially-averaged energy computation on each component of the motion fields and summing them.

TUPANN has similarly lacking high-frequency performance to PySTEPS and EarthFormer, when compared to models whose focus is obtaining sharp predictions (CasCast and NowcastNet). On the other hand, its low frequencies perform well over all considered time horizons.

Perhaps surprisingly, the power spectra of the motion fields produced by TUPANN and NowcastNet share a similar profile, contradicting our visual analysis of NowcastNet's motion fields. We observe that DARTS' motion fields are truncated for moderate to high frequencies, which is an intended consequence of its underlying algorithm. The VED seems to reproduce the low frequencies of DARTS closely, but this alignment decays as the lags increase. The VED does not learn the truncation of high frequencies from DARTS.

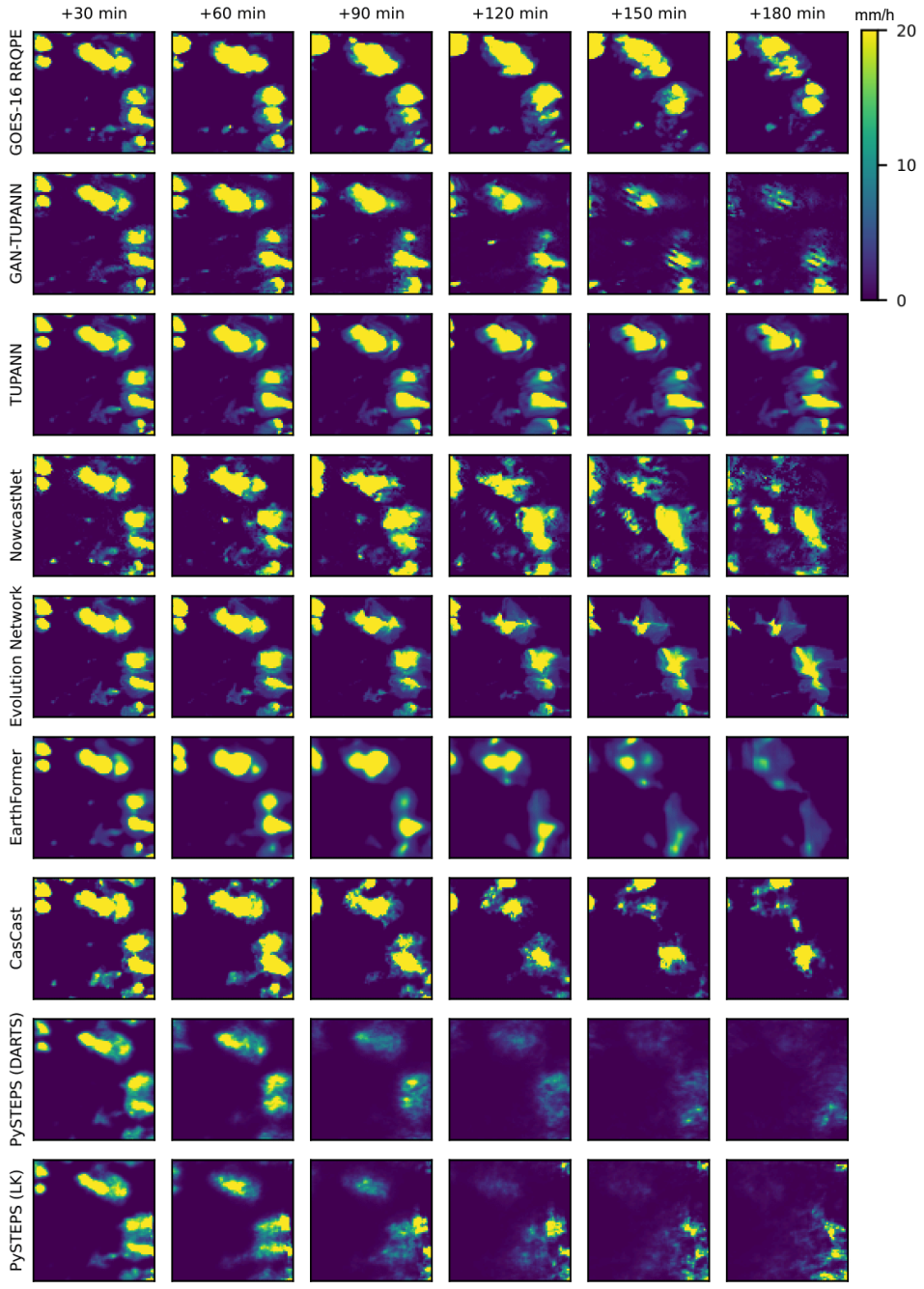

Figure 6: Predictions up to 3h ahead for a rain event in Manaus, starting at 2021-11-20 22:00 UTC. Rows show the ground truth, TUPANN, and other models; columns represent lead times. TUPANN displays greater skill in predicting the movement and intensity of the rain event, while generative models produce visually sharper but less accurate predictions.

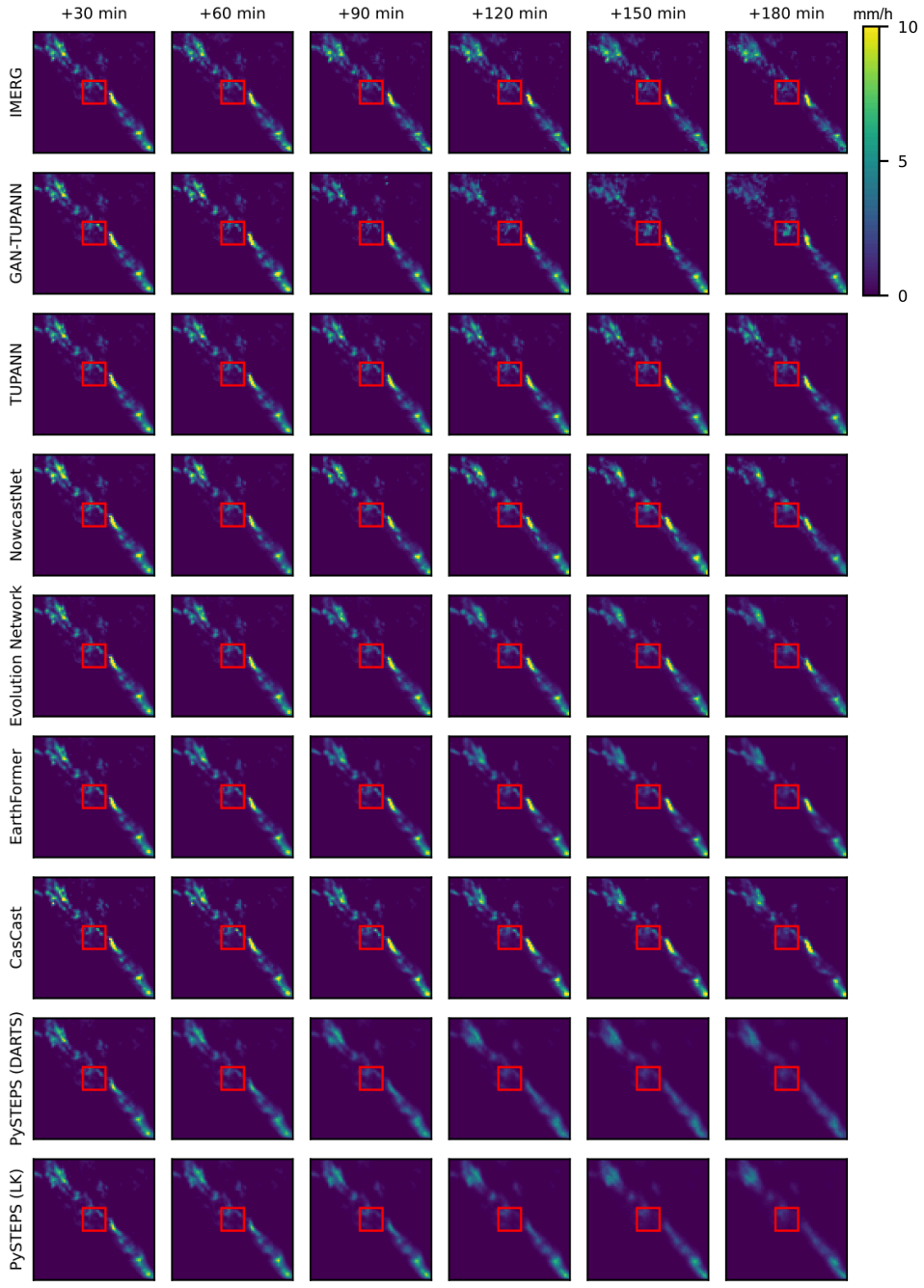

Figure 7: Prediction sample from the IMERG test dataset centered in Rio de Janeiro, starting from 2023-01-08 07:30 UTC. The red square represents the same area as in the GOES-16 figures, demonstrating the difference in spatial resolution.

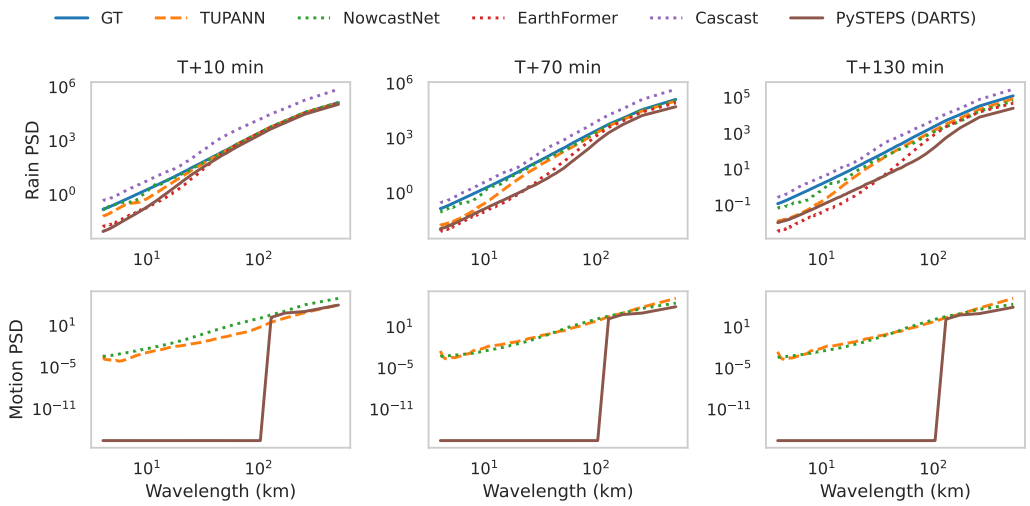

Figure 8: Radially-averaged energy of Fourier coefficients versus wavelength. Columns represent different time horizons. The first row exhibits the usual PSD metric while the second provides an analogue computed on the motion fields associated to the respective model. TUPANN presents good alignment with the ground truth under low frequencies but lacks high-frequency refinement.

C Ablation

C.1 Latent Space Rollout

We test a simpler approach to the model architecture in which the optical flow learning step is completely removed. The model is simply composed of a Variational Encoder Decoder that takes a past sequence of images as input and predicts the most recent image. A Visual Transformer is used to predict future latent space representations by receiving an initial latent representation and a number indicating the desired lead time. This simplified architecture is illustrated in Figure 9. Table 5 shows the performance of this version compared to the one proposed in the main text, evaluated in the city of Rio de Janeiro. It is clear that the original version is superior in terms of all proposed CSI thresholds. Figure Figure 10 illustrates the dependence of the mean CSI value on lead time, again making explicit the performance loss. This illustrates the importance of regularizing the training process through the guidance of underlying learned motion and intensity fields.

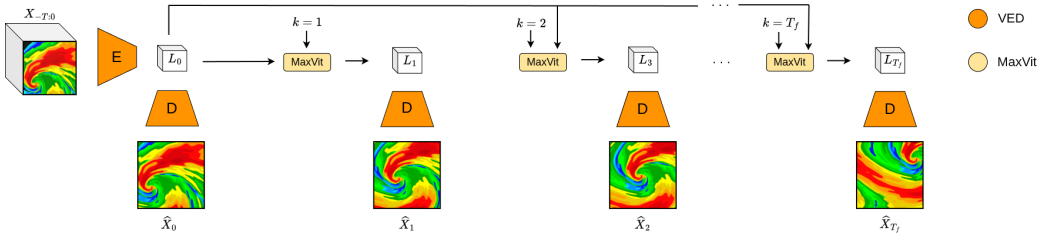

Figure 9: Simplified TUPANN architecture. Learning of motion and intensity fields is removed, remaining only an Encoder-Decoder and a Visual Transformer trained conditionally on the lead time.

C.2 End-to-end supervision

We provide the results of an alternative learning scheme, suggested by an anonymous reviewer: we dispense the optical flow supervision, computing the VED loss directly on the final extrapolated images. The Visual Transformer component of the model is kept unchanged. We nominate this

Table 5: Aggregated CSI metrics for Rio de Janeiro with GOES-16 data comparing original architecture and a simplified version without learning of motion and intensity fields. Bold denotes the best model.

Model	CSI-M ↑		CSI_4 ↑		CSI_8 ↑		CSI_{16} ↑		CSI_{32} ↑		CSI_{64} ↑	
	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4	POOL1	POOL4
TUPANN	**0.259**	**0.277**	**0.330**	**0.384**	**0.289**	**0.289**	**0.330**	**0.336**	**0.274**	**0.287**	**0.072**	**0.090**
Simplified TUPANN	0.205	0.189	0.270	0.270	0.237	0.192	0.257	0.243	0.203	0.199	0.056	0.043

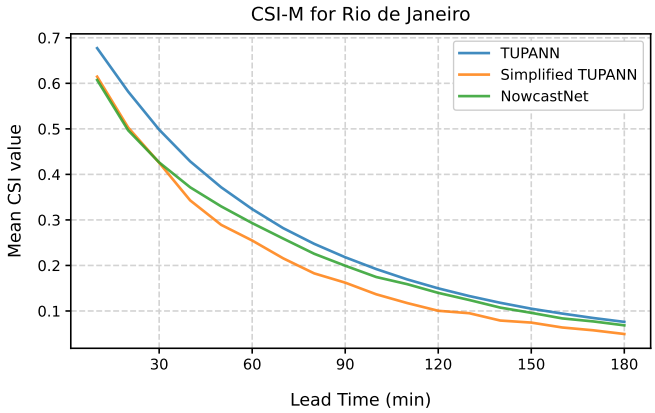

Figure 10: Mean CSI versus lead time for GOES-16 data in Rio de Janeiro comparing original architecture and a simplified version without learning of motion and intensity fields.

modified version TUPANN-U (TUPANN Unsupervised by optical flow). To this end, we combine our approach with the above mentioned warp function for training, as originally proposed in Zhang et al. (2023). The architecture of the VED is not changed; it still predicts a motion and an intensity field. However, unlike our initial proposal, after the prediction of the VED is made, the warp function is used to extrapolate the most recent image to the next temporally using the predicted motion and intensity fields. This allows for training the encoder-decoder directly on the image space, without the need of optical flow supervision. More explicitly, the new training loss is given by:

$$\text{Loss}_{\text{VED}} = \ell_1 \left(\text{warp} \left(\widehat{v}_{0\to 1}, \widehat{s}_{0\to 1}, X_0 \right), X_1 \right),$$

where $\widehat{v}_{0\to 1}, \widehat{s}_{0\to 1}$ are the outputs of the autoencoder. Figure Figure 11 illustrates a performance gain uniformly across lead times in terms of mean CSI for this modified training scheme. In Figure 12, on the other hand, a comparison is drawn between the motion fields predicted by this new approach, the ones predicted by the original approach, and the ones obtained with the DARTS optical flow method. The motion fields learned by training the VED with a modified loss present a perceptible degradation with respect to the ones produced by the optical-flow supervised VED, leading to unrealistic motion.

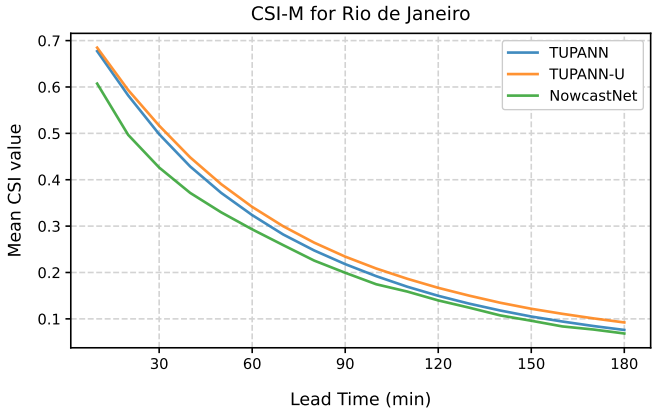

Figure 11: Mean CSI versus lead time for GOES-16 data in Rio de Janeiro comparing original architecture and a modified version using the warp function to compute VED loss directly on image space (TUPANN-U).

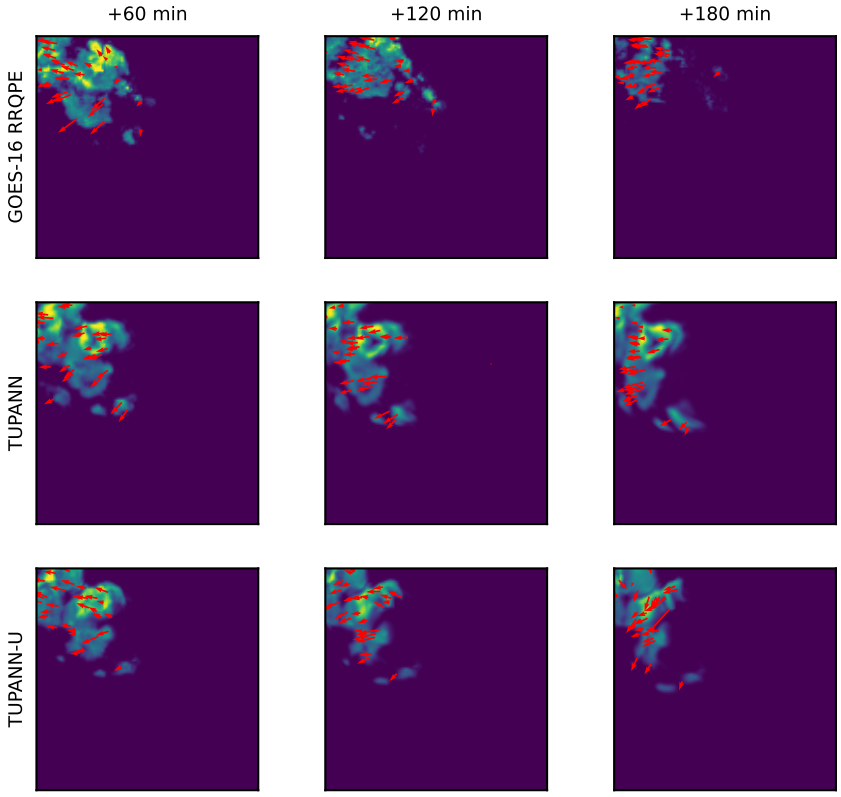

Figure 12: Prediction sample from the GOES-16 RRQPE test dataset centered in Rio de Janeiro, starting from 2023-12-06 04:40 UTC. We compare TUPANN with a modified version that circumvents optical flow supervision (TUPANN-U). As expected, motion field smoothness is sacrificed for accuracy.