Data-driven Precipitation Nowcasting Using Satellite Imagery

Supplementary Material

In this Supplementary material, we provide the following details:

- Architecture details of neural precipitation model (NPM).
- Detailed distribution information about the Sat2Rdr dataset.
- Additional qualitative and quantitative results.

Implementation Details

In this section, we provide a detailed description of NPM's architecture, including implementation specifics that are not included in the main paper. NPM consists of a Satellite Prediction model and a Satellite-to-Radar model.

Satellite Prediction Model

Table 1 details the architecture of the Satellite Prediction Model used in NPM, which is based on the Video Prediction baseline. The optimal number of translators, encoders, and decoders may vary depending on the dataset.

Time Embedding			
Name	**Layer**	**Input Shape**	**Output Shape**
Input	-	[b, 1]	[b, 1]
Time Embedding*	SinusoidalPosEmb	[b, 1]	[b, 64]
	Linear	[b, 64]	[b, 256]
	GeLU	[b, 256]	[b, 256]
	Linear	[b, 256]	[b, 64]
	GeLU	[b, 64]	[b, 64]
	Linear	[b, 64]	[b, 640]
Reshape	Reshape	[b, 64 × 6]	[b, 64 × 6, 1, 1]
Satellite Prediction Model			
Name	**Layer**	**Input Shape**	**Output Shape**
Input	-	[b, 6, 4, 768, 768]	[b, 6, 4, 768, 768]
Reshape	Reshape	[b ×6 , 4, 768, 768]	[b ×6 , 4, 768, 768]
Encoder** × 4	Conv2d (kernel=3, stride=[1, 2, 1, 2])	[b × 6, 4, 768, 768]	-
	LayerNorm	-	-
	SiLU	-	[b × 6, 64, 48, 48]
Reshape	Reshape	[b × 6, 64, 16, 16]	[b, 64 ×6, 48, 48]
Translator × 6	Conv2d (kernel=7, padding=3, group=4)	[b, 64 ×6, 48, 48]	[b, 64 ×6, 48, 48]
	Add Time Embeddings*	[b, 64 ×6, 48, 48]	[b, 64 ×6, 48, 48]
	GeLU	[b, 64 ×6, 48, 48]	[b, 64 ×6, 48, 48]
	Conv(kernel=1)	[b, 64 ×6, 48, 48]	[b, 64 ×6, 48, 48]
	GeLU	[b, 64 ×6, 48, 48]	[b, 64 ×6, 48, 48]
	Conv(kernel=1)	[b, 64 ×6, 48, 48]	[b, 64 ×6, 48, 48]
	Add Spatio-temporal Attention	[b, 64 ×6, 48, 48]	[b, 64 ×6, 48, 48]
Reshape	Reshape	[b, 64 ×6, 48, 48]	[b ×6, 64, 48, 48]
Decoder × 4	Pixel Shuffle (kernel=3, stride=1, upscale_factor=[2,1,2])	[b ×6, 64, 48, 48]	-
	LayerNorm	-	-
	SiLU	-	[b ×6, 64, 768, 768]
	Conv2d (kernel=1)	[b ×6, 64, 768, 768]	[b ×6, 4, 768, 768]
Reshape	Reshape	[b ×6, 4, 768, 768]	[b, 6, 4, 768, 768]
Output	-	[b, 6, 4, 768, 768]	[b, 6, 4, 768, 768]

Table 1: An architecture of Satellite Prediction Model

Satellite-to-Radar Model

Table 2 provides an overview of the architecture for the Satellite-to-Radar Model utilized in NPM, which employs an image-to-image translation model. Our baseline is based on StegoGAN (Wu et al. 2024) in the paired setting. StegoGAN comprises ResNet (He et al. 2016)-based generators and PatchGAN (Isola et al. 2017)-based discriminators. To implement StegoGAN, we use the official repository.

Satellite-to-Radar Model			
Name	**Layer**	**Input Shape**	**Output Shape**
Input	-	[b, 3, 768, 768]	[b, 3, 768, 768]
Generator × 2	Conv2D (kernel=7, stride=1)	[b, 3, 768, 768]	[b, 64, 768, 768]
	ReLU	[b, 64, 768, 768]	[b, 64, 768, 768]
	Conv2D (kernel=3, stride=2)	[b, 64, 768, 768]	[b, 128, 384, 384]
	ReLU	[b, 128, 384, 384]	[b, 128, 384, 384]
	Conv2D (kernel=3, stride=2)	[b, 128, 384, 384]	[b, 256, 192, 192]
	ReLU	[b, 256, 192, 192]	[b, 256, 192, 192]
	Residual Block × 6 (kernel=3, stride=1)	[b, 256, 192, 192]	[b, 256, 192, 192]
	Conv2D Transpose (kernel=3, stride=2)	[b, 256, 192, 192]	[b, 128, 384, 384]
	ReLU	[b, 128, 384, 384]	[b, 128, 384, 384]
	Conv2D Transpose (kernel=3, stride=2)	[b, 128, 384, 384]	[b, 64, 768, 768]
	ReLU	[b, 64, 768, 768]	[b, 64, 768, 768]
	Conv2D (kernel=7, stride=1)	[b, 64, 768, 768]	[b, 1, 768, 768]
	Tanh	[b, 1, 768, 768]	[b, 1, 768, 768]
Discriminator × 2	Conv2D (kernel=4, stride=2)	[b, 1, 768, 768]	[b, 64, 384, 384]
	LeakyReLU	[b, 64, 384, 384]	[b, 64, 384, 384]
	Conv2D (kernel=4, stride=2)	[b, 64, 384, 384]	[b, 128, 192, 192]
	LeakyReLU	[b, 128, 192, 192]	[b, 128, 192, 192]
	Conv2D (kernel=4, stride=2)	[b, 128, 192, 192]	[b, 256, 96, 96]
	LeakyReLU	[b, 256, 96, 96]	[b, 256, 96, 96]
	Conv2D (kernel=4, stride=1)	[b, 256, 96, 96]	[b, 512, 95, 95]
	LeakyReLU	[b, 512, 95, 95]	[b, 512, 95, 95]
	Conv2D (kernel=4, stride=1)	[b, 512, 95, 95]	[b, 1, 94, 94]

Table 2: An architecture of Satellite-to-Radar model

Dataset Distribution

Fig. 1 shows the monthly rainfall distribution of radar data in the Sat2Rdr dataset. As seen in the figure. During the summer (June to August), the monsoon season begins across the Korean Peninsula, increasing the overall proportion of heavy rain (rain rate $\geq$ 8 mm/hr). Conversely, the proportion of heavy rain decreases in the colder and drier months of December and January.

Additionally, since the majority of precipitation falls within the light rain category (rain rate $\leq$ 1 mm/hr), it seems necessary to explore ways to improve the prediction of heavy rainfall.

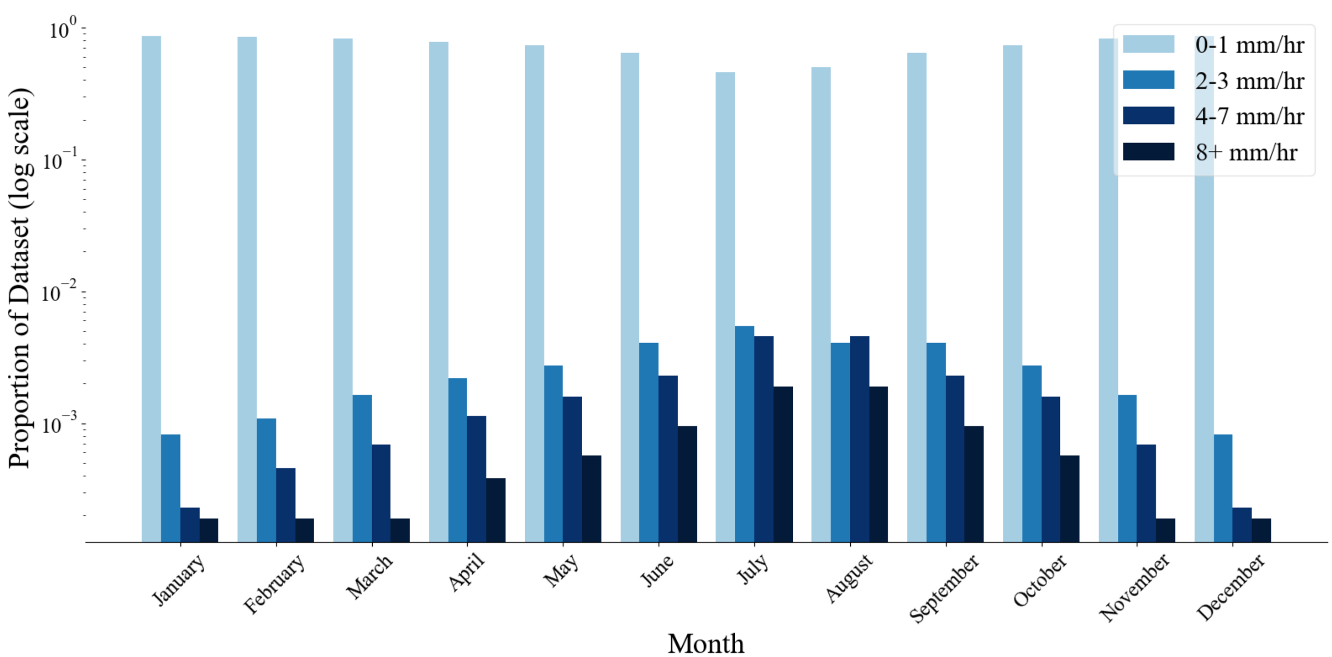


Figure 1: Seasonal rainfall distribution statistics of the Sat2Rdr dataset

Additional Results

Quantitative Results

Table 3 presents additional quantitative results, including the probability of detection (POD) and false alarm ratio (FAR) metrics, for various video prediction models. Consistent with the trends observed in comparing CSI performance in the main paper, performance declines as lead time increases. Our proposed framework demonstrates the highest performance in both POD and FAR metrics.

	POD 1 mm ↑						FAR 1 mm ↑					
Method	1h	2h	3h	4h	5h	6h	1h	2h	3h	4h	5h	6h
PhyDNet (Guen and Thome 2020)	0.82	0.75	0.67	0.60	0.55	0.50	0.21	0.26	0.32	0.38	0.44	0.50
PredRNNV2 (Wang et al. 2022)	0.85	0.77	0.69	0.62	0.56	0.52	0.20	0.25	0.31	0.37	0.43	0.48
SimVP (Gao et al. 2022)	0.80	0.73	0.65	0.58	0.52	0.48	0.23	0.28	0.34	0.40	0.46	0.51
SimVP-V2 (Tan et al. 2022)	0.78	0.71	0.64	0.56	0.50	0.46	0.24	0.29	0.35	0.41	0.47	0.52
TAU (Tan et al. 2023)	0.81	0.74	0.66	0.59	0.54	0.49	0.22	0.27	0.33	0.39	0.45	0.51
SwinLSTM (Tang et al. 2023)	0.83	0.76	0.68	0.61	0.55	0.51	0.21	0.26	0.32	0.38	0.44	0.50
Ours	**0.90**	**0.82**	**0.74**	**0.68**	**0.63**	**0.59**	**0.18**	**0.23**	**0.28**	**0.33**	**0.39**	**0.45**

Table 3: Comparison of POD and FAR performance between Video Frame Prediction models and our model.

Qualitative Results

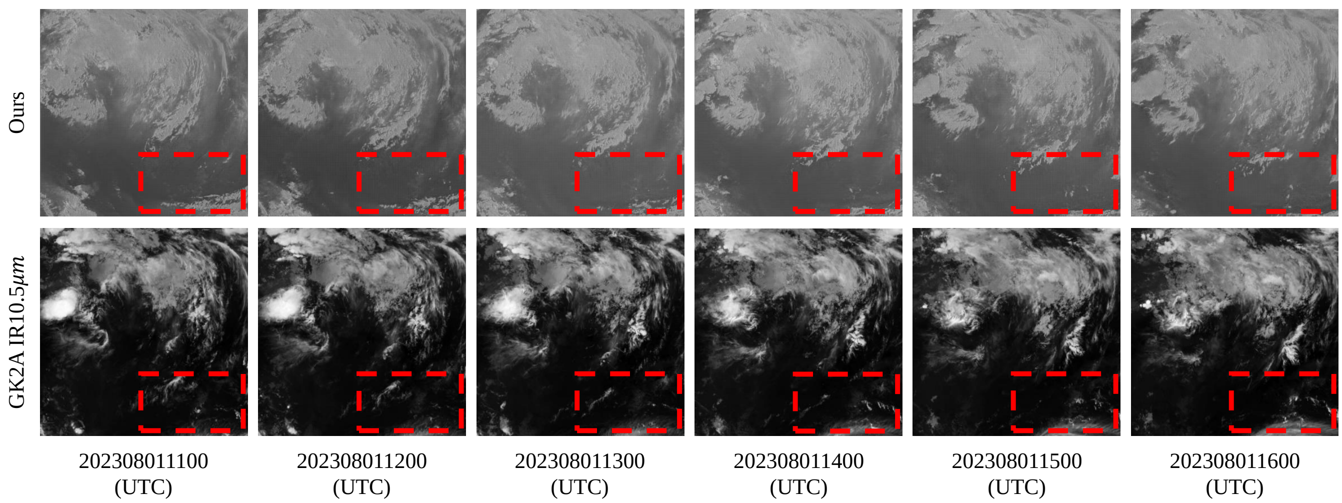
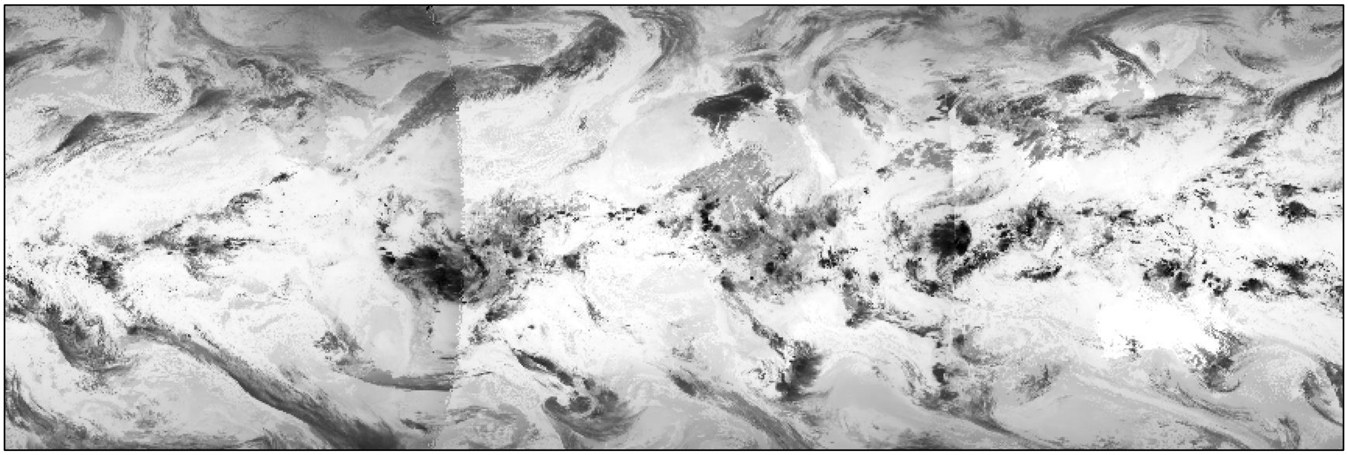


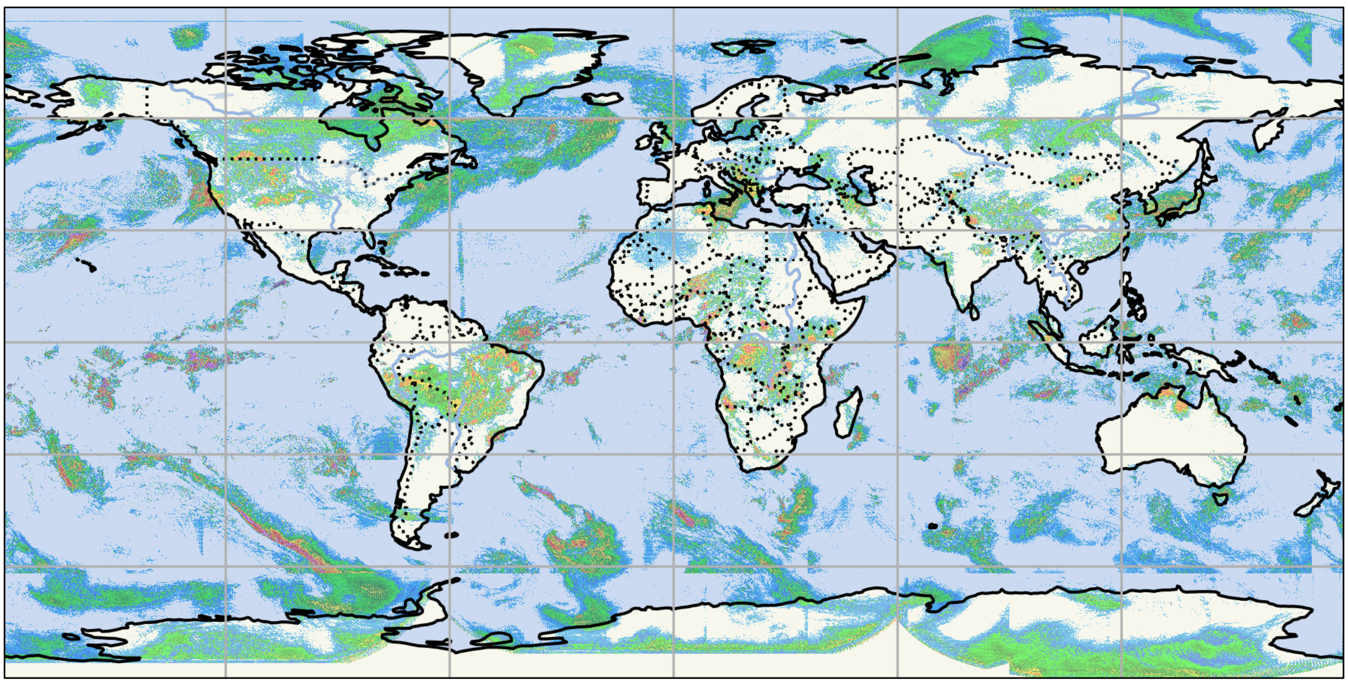
Figure 2: Qualitative results from NPM

Fig. 2 shows the qualitative results of the video prediction models. Our model predicts satellite imagery for the next six steps by considering seasonality and the movement of clouds. While our model generally predicts the overall direction and flow well, the model struggles with signals entering from the borders, as highlighted by the red boxes in the figure.

To address this issue, we plan to resolve the border problem by aligning and predicting using global satellite data. Fig. 3 illustrates the results of globally aligning geostationary satellites to create satellite imagery that covers the entire Earth, followed by satellite-to-radar translation. As shown in the figure, training on data from the entire globe would eliminate borders, potentially solving the border issue. However, aligning global data significantly increases the image size, making the development of efficient learning methods necessary.



(a) Our geostationary ring IR10.8 μm image (GEOS-18, MSG4, GK2A)



(b) Our satellite-to-radar results

Figure 3: An overview of precipitation estimation using global satellite images at 2 km intervals.

References

Gao, Z.; Tan, C.; Wu, L.; and Li, S. Z. 2022. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guen, V. L.; and Thome, N. 2020. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tan, C.; Gao, Z.; Li, S.; and Li, S. Z. 2022. SimVP: Towards Simple yet Powerful Spatiotemporal Predictive Learning. *arXiv preprint arXiv:2211.12509*.

Tan, C.; Gao, Z.; Wu, L.; Xu, Y.; Xia, J.; Li, S.; and Li, S. Z. 2023. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tang, S.; Li, C.; Zhang, P.; and Tang, R. 2023. Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wang, Y.; Wu, H.; Zhang, J.; Gao, Z.; Wang, J.; Philip, S. Y.; and Long, M. 2022. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(2): 2208–2225.

Wu, S.; Chen, Y.; Mermet, S.; Hurni, L.; Schindler, K.; Gonthier, N.; and Landrieu, L. 2024. StegoGAN: Leveraging Steganography for Non-Bijective Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.