Probabilistic Precipitation Nowcasting with Rectified Flow Transformers

Supplementary Material

A. Extended Results

A.1. Extreme Precipitation

We further evaluate our method under extreme weather conditions. We define extreme weather as the top 20 % of events with the highest average precipitation, and the rest as "normal".

Forecasting Performance We provide a quantitative comparison between our method and the state-of-the-art CasCast method [40] in extreme weather in Tab. 6. Under extreme conditions, the CRPS and SSIM deteriorate for all models, indicating that prediction is more difficult in chaotic extreme events. Interestingly, localization-centric metrics improve for all methods. Therefore, determining the movement of extreme patterns seems to be easier than predicting the onset of light to medium rain. Notably, the performance gap for these metrics between our method and CasCast further narrows. Hence, our approach achieves comparable localization performance to state-of-the-art methods in critical extreme weather scenarios while maintaining superior distribution coverage. Furthermore, the scaling properties discussed in Sec. 4.3 hold under extreme conditions. The L-LSM achieves the best CRPS and SSIM scores, and HSS and CSI match the results from CasCast [40] closely with classifier-free guidance. Therefore, our L variant is particularly well-suited to determine localized risk at improved distribution coverage compared to prior methods.

Reconstruction Performance In addition to the regression experiment in Fig. 6 in the main paper, we quantitatively analyze the differences between reconstruction ensemble members for extreme and normal weather in Tab. 7. We find $2.4\times$ higher variance with extreme weather, and intra-ensemble differences, as measured by RMSE, are $1.5\times$ higher for extreme weather. This result confirms that reconstructions of chaotic extreme weather events show larger ensemble differences, indicating higher uncertainty.

Table 6. Comparison of CasCast [40] with our method for extreme weather events. Extreme weather is defined as the 80th percentile of all events with the highest precipitation.

Model	CRPS↓	SSIM↑	HSS↑	CSI↑
CasCast [40]	0.0404	0.6354	**0.5726**	**0.4728**
Our B-LSM	0.0384	0.6514	0.4982	0.4162
↳ with cfg	0.0400	0.6634	0.5321	0.4678
Our L-LSM	**0.0357**	0.6602	0.5107	0.4233
↳ with cfg	0.0360	**0.6731**	0.5659	0.4677

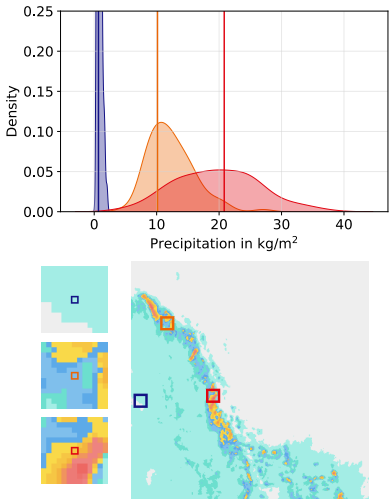


Figure S1. Forecasting ensemble distributions for three pixels of low, medium, and high precipitation regions. Vertical lines show ground truth precipitation. The more extreme the weather, the larger the distribution spread, indicating higher uncertainty.

Table 7. Differences over reconstructions from a 10-member *T-reg.* FREUD ensemble. Extreme weather events are defined as the 80th percentile of all events in the validation dataset with the highest average precipitation.

Weather Data	Var $\times 10^{-5}$	SD $\times 10^{-3}$	RMSE $\times 10^{-2}$	MAE $\times 10^{-3}$
Extreme	6.866	5.066	1.172	5.737
Normal	2.849	2.003	0.769	2.884

Extreme Weather Phenomena We further assess how well our method generalizes to rare and extreme weather phenomena by evaluating the few labeled severe events in the SEVIR train and test sets. As shown in Tab. 8, our approach consistently outperforms CasCast across tornado, flood, and flash-flood cases, demonstrating clear advantages. Beyond these quantitative results, we show a qualitative example in Fig. S2, which shows that our model can capture the large-scale rotational dynamics and global motion patterns characteristic of a hurricane. We further provide video visualizations that illustrate the circular motion more clearly.

A.2. Ensemble Distributions

Error Distributions Fig. S4 shows the distribution of deviations from ground-truth precipitation, for FREUD and our forecasting model (B-LSM), where 0 denotes a perfect forecast. We only consider pixels where precipitation is ob-

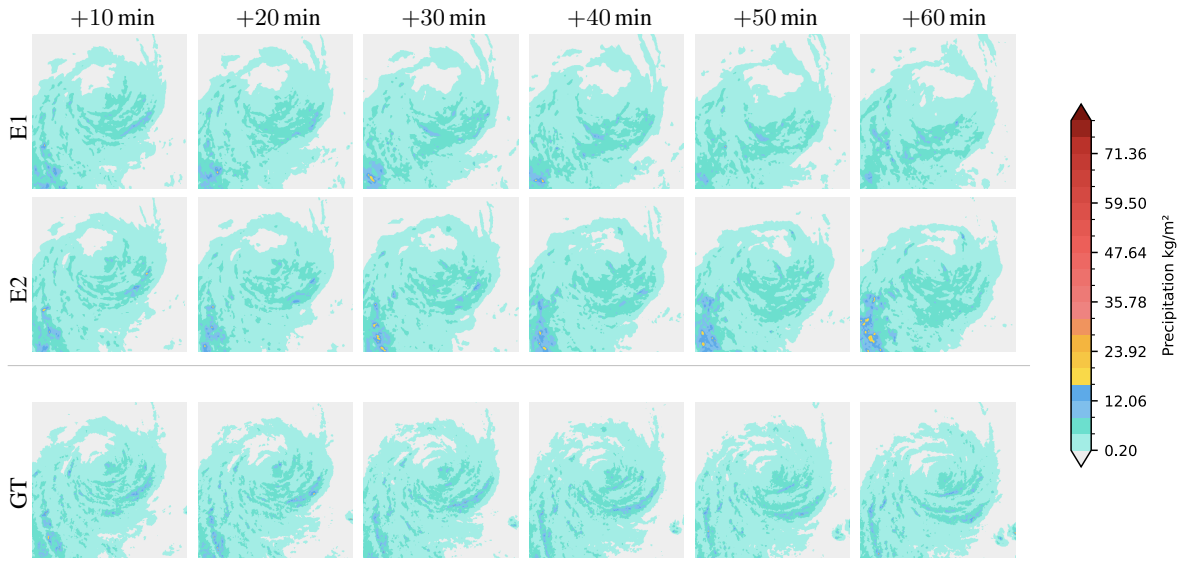


Figure S2. Qualitative sample from a known hurricane event. Our method is able to capture the circular motion observed in the ground truth. For a better visualization of circular motion, please refer to the supplemented video visualizations.

Table 8. Reconstruction and forecasting performance on rare and severe SEVIR weather events. We compare CasCast [40] with our method across reconstruction metrics (RMSE, PSNR, SSIM) and forecasting metrics (CRPS, HSS) for tornado, flood, and flash-flood subsets.

		Reconstruction			Forecast	
Event	**Model**	RMSE	PSNR	SSIM	CRPS	HSS
Tornado (194 events)	CasCast	0.104	22.800	0.957	0.0627	0.4675
	Ours	0.0120 −0.092	38.434 +15.634	0.970 +0.013	0.0374 −0.0253	0.4882 +0.0207
Flood (177 events)	CasCast	0.080	25.316	0.960	0.0488	0.4493
	Ours	0.011 −0.069	39.236 +13.920	0.971 +0.011	0.0280 −0.0208	0.4106 −0.0387
Flash Flood (385 events)	CasCast	0.072	25.619	0.961	0.0545	0.4660
	Ours	0.012 −0.060	38.378 +12.759	0.971 +0.010	0.0307 −0.0238	0.5140 +0.048

served. Both produce nearly zero-centered error distributions with similar likelihoods of over- and underestimation. The figure reflects a slight tendency to underestimate high-precipitation regions, consistent with average weather conditions; however, it overall indicates that ensemble members are well-spread around the ground truth.

Ensemble Performance Fig. S3 shows the per ensemble member MAE as well as the corresponding ensemble CRPS and MAE of the mean over ensemble members for the qualitative sample of Fig. 4 in the main paper. Across all lead times, the ensemble consistently outperforms the average performance of its individual members. This indicates that individual errors tend to spread around a common mean, and discrepancies cancel out when aggregated, yielding a mean forecast that closely aligns with the ground truth. This behavior is consistent with the error distributions discussed earlier.

Pixel Value Distributions Fig. S1 shows the distribution of LSM ensemble forecasts for pixels in low, medium, and high precipitation regimes. The mode shifts with the true intensity, and the variance increases for heavier rainfall, reflecting higher uncertainty in chaotic, high-precipitation re-

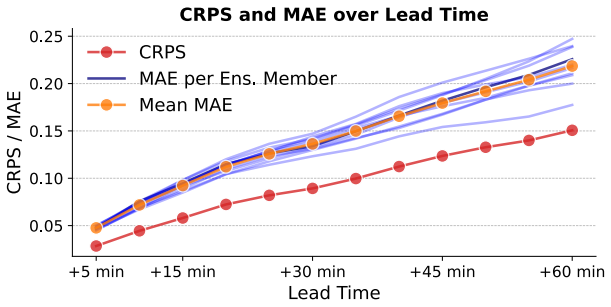


Figure S3. Forecast skill of the individual ensemble members and of their aggregated prediction for the example in Fig. 4. Each member constitutes a plausible realisation of the future weather, yet the expectation over ensemble members attains lower error.

gions. All regimes exhibit a long tail toward larger values, capturing the possibility of intensifying rain; this tail becomes more pronounced for high precipitation. In this regime, however, the ensemble mean tends to underestimate the true value, consistent with climatology, where a decrease in intensity is more common than further escalation. Overall, the ensemble behavior aligns with known precipitation dynamics, indicating that our model has learned realistic weather statistics.

A.3. Performance on MeteoNet

In addition to the results obtained for the SEVIR [139] benchmark, we validate our model's applicability to other datasets by applying our model to the MeteoNet [70] benchmark. Tab. 9 shows that our model performs similarly on MeteoNet and SEVIR. Further, our model outperforms all baselines but CasCast in terms of CRPS and achieves comparable CSI. However, we find that results on MeteoNet are highly sensitive to the chosen train–test split. Fig. S5 shows statistical differences between MeteoNet training splits. While differences are small, the date-based split shows a slightly higher mean, resulting in lower CSI for the random split as true negatives have no effect on CSI computation. Since CasCast [40] provides limited details on their experimental setup and no public MeteoNet checkpoint, we cannot fully validate comparability under their protocol. For this reason, we focus our main analysis on SEVIR.

Table 9. **Performance on MeteoNet:** Our approach achieves competitive performance on MeteoNet and performs similarly to the model trained on the SEVIR dataset. Yet, we observe a strong influence of the train-test split on downstream performance.

Method	Split	CRPS↓	SSIM↑	HSS↑	CSI↑
EarthFormer [35] (NeurIPS '22)	unknown	0.0224	–	–	0.2831
NowcastNet [158] (Nature '23)	unknown	0.0277	–	–	0.2955
PreDiff [37] (NeurIPS '23)	unknown	0.0197	–	–	0.2546
CasCast [40] (ICML '24)	unknown	0.0180	–	–	0.3156
FREUD + LSM-L (*ours*)	Random	0.0224	0.7212	0.0876	0.1117
↳ with cfg		0.0231	0.7133	0.1368	0.0876
FREUD + LSM-L (*ours*)	Date-based	0.0193	0.7312	0.2082	0.1417
↳ with cfg		0.0194	0.7405	0.3150	0.2191

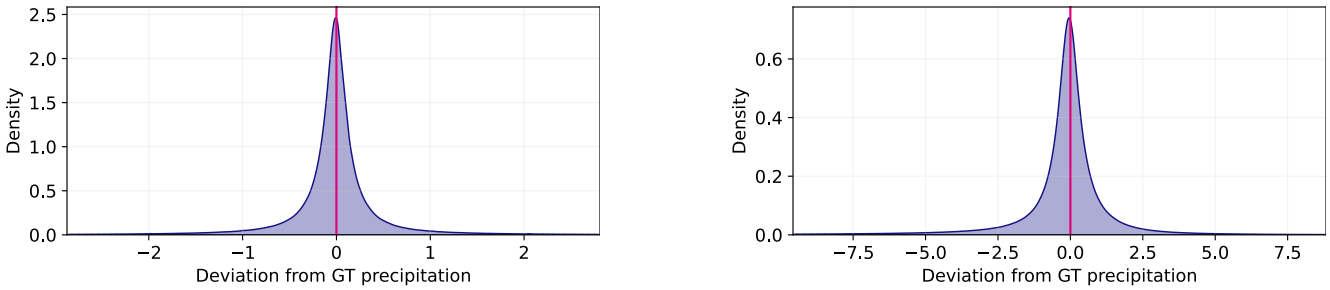
A.4. Effect of Classifier-free Guidance

Impact on Performance We analyze the impact of classifier-free guidance (CFG) [48] on forecasting performance in Fig. S6, comparing our B-LSM to the state-of-the-art CasCast method [40]. As guidance strength increases, CRPS rises monotonically, reflecting reduced distributional coverage, consistent with observations in image generation [48, 93]. HSS, however, improves up to an optimal guidance level (1.5 for our model) before declining. We observe the same behavior with Adaptive Projected Guidance (APG) [112], despite its design to counteract over-saturation in image synthesis at high guidance strengths.

Qualitative Effect We visualize two qualitative samples at different guidance scales with our method (Fig. S15 and Fig. S16) and CasCast (Fig. S17 and Fig. S18). For both methods, we observe that the extremeness of weather conditions increases with higher guidance. Samples degenerate into unrealistic extreme weather beyond a method-specific threshold. As APG does not solve this problem, we hypothesize that this behavior is distinct from over-saturation. Analyzing unconditional samples provided in Fig. S19, we find notably low precipitation intensity, as low precipitation is more common than strong rain. Hence, by applying guidance, we push samples from low to high precipitation.

Impact on Descriptive Statistics Inspired by these qualitative insights, we investigate the descriptive statistics of forecasts with different guidance scales in Fig. S7. For our method, we find that guidance consistently increases the average precipitation intensity in forecasts. For CasCast, we observe a different yet related behavior: while the mean intensity remains largely unaffected, extreme values increase dramatically, leading to unrealistically high precipitation that quickly exceeds the strongest precipitation found in the SEVIR dataset [139]. Therefore, both methods produce samples with unrealistically high precipitation with strong guidance, which is likely not an over-saturation artefact.

Unsuitability of CFG for Forecasting Based on our previous findings, we believe the problem with guidance in generative precipitation nowcasting is related to the dominance of low precipitation in the datasets. Generative models that learn the data distribution will tend to produce samples with low precipitation, resulting in unconditional samples with less intense precipitation than conditional samples. Therefore, when applying classifier-free guidance, we push samples towards higher precipitation, which leads to the observed unrealistic samples. Therefore, guidance is flawed in nowcasting, as the potential improved alignment with conditions is confounded by a distribution shift. Furthermore, guidance induces reduced distribution coverage and stronger overconfidence, which is problematic for uncertainty-aware forecasting of chaotic weather systems. This result is not limited to our method and applies to state-of-the-art approaches, such as CasCast [40], as well. Therefore, it is a general flaw of diffusion-based nowcasting. Notably, our method performs well without guidance, achieving superior distribution coverage and perceptual similarity at competitive localization compared to methods that rely on guidance (see Tab. 2 in main paper).



(a) FREUD (b) B-LSM

Figure S4. Deviations from ground truth precipitation, whereby 0 denotes a perfect forecast. Only pixels with more than $1\,kg/m^2$ precipitation are considered. The x-axis range indicates three standard deviations from the mean to remove outliers. Both distributions are almost zero-centered and show similar likelihoods of over- and underestimating precipitation.

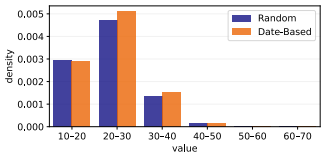
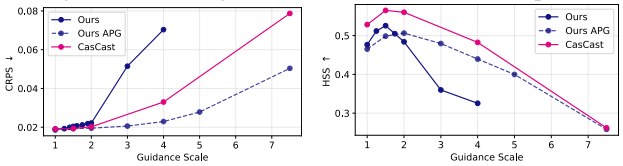


Figure S5. Training data distribution of MeteoNet splits.



(a) CRPS (b) HSS

Figure S6. Performance of our B-LSM and CasCast with increasing CFG [48] and APG [112] guidance strength. For all approaches, CRPS continuously worsens while HSS improves up to an optimal value and deteriorates afterwards.

A.5. Further Ablations

Latent Space Distributions Fig. S8 shows the density distribution of the latent space for all regularization variants. Without regularization, we observe high norms and variance, resulting in low density and long tails. Moreover, we observe a bimodal distribution for all latent channels. The same bimodal pattern is observed with *KL-reg.* FREUD, but the variance is substantially reduced, leading to a higher density throughout the value range, and the distribution is almost zero-centered. However, we still observe long tails in the distributions. Similarly, the CasCast latent value distribution exhibits reduced variance and an almost zero-centered latent space; however, we still observe very long tails due to the weak KL regularization. In comparison, the *T-reg.* latent space exhibits the most zero-centered distribution with the lowest variance. We do not observe heavy tails and a less pronounced bimodal pattern; thus, the density remains high across the $[-1, 1]$ value range. This indicates that the *T-reg.* FREUD encoder produces a latent distribution that is easier to learn and sample for both the generative decoder and downstream latent space model (see Tab. 5 in main paper).

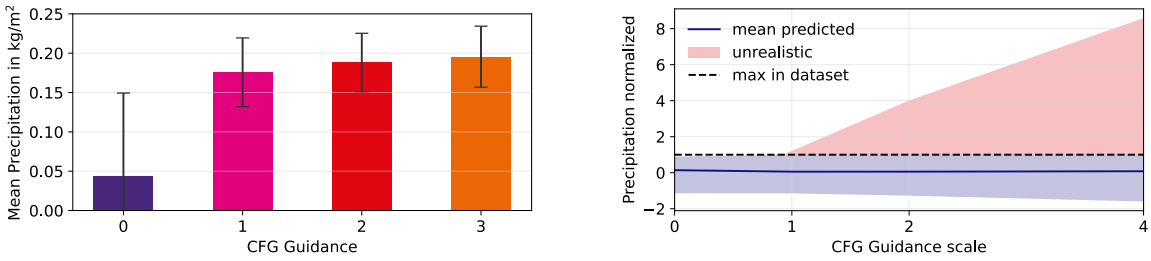
Table 10. Comparison of forecasting skill with our B-LSM using factorized spatio-temporal attention and full self-attention. We find slightly improved performance with the full attention variant.

Attention	CRPS↓	SSIM↑	HSS↑	CSI↑
Factorized	0.0196	0.7828	0.4923	0.3805
Full	0.0187	0.7897	0.4968	0.3848

Number of Conditioning Frames The masking-based training paradigm (RaMViD) [54] allows variable conditioning frames during inference. We find improved performance when using more conditioning frames in Fig. S9.

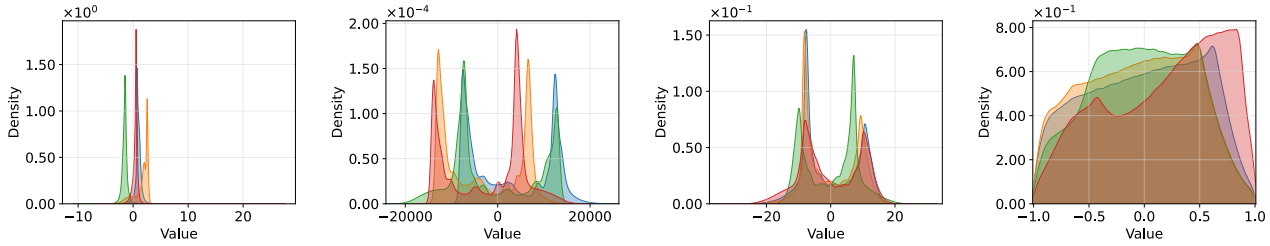
Full vs Factorized Attention We further train a latent space model with full attention over space and time, and compare it to our spatio-temporally factorized attention model in Tab. 10. Full attention provides slightly better performance, but its computational cost is prohibitive: The complexity of full self-attention is given by $(T \times H \times W)^2$, whereas the complexity of factorized attention is given as $(H \times W)^2 + T^2$. For our use case, we have $T = 25$, $H = 24$, and $W = 24$ after latent embedding and patching (see Sec. B), therefore, we require 207.4 GFLOPs for full attention and 0.3 GFLOPs for factorized attention. We believe that for operational nowcasting, such marginal improvements do not justify the substantial computational overhead; therefore, we only report results from the factorized model throughout the paper.

Blob Toy Experiment We emulate an experiment from prior work [14] to evaluate uncertainty quantification by inserting areas of unmoving extreme precipitation (blobs) into the data and assessing the variance of reconstruction ensembles. Fig. S10 shows a linear regression of the number of blobs against the ensemble variance. We find a significant linear correlation between ensemble variance and the blob count, confirming that reconstruction variance can de-



(a) Ours (b) CasCast

Figure S7. The effect of cfg [48] on our method and CasCast [40]. For our approach (a), the mean precipitation consistently increases while the ensemble variance (error bars) decreases. For CasCast (b), the mean remains largely unaffected, but the min-max range of predicted values (shaded area) explodes. We keep the encoded $[0, 1]$ range in the right plot instead of applying the non-linear mapping to precipitation for visualization.



(a) CasCast $M = 0.367$, $\sigma = 1.481$

(b) Unreg. FREUD $M = -741.841$, $\sigma > 1 \times 10^3$

(c) KL-reg. FREUD $M = 0.040$, $\sigma = 8.779$

(d) T-reg. FREUD $M = 0.0261$, $\sigma = 0.499$

Figure S8. Latent space distributions for different regularization schemes and the CasCast encoder [40]. Colors indicate the latent space channel. The x-axis range denotes the min-max range of latent values.

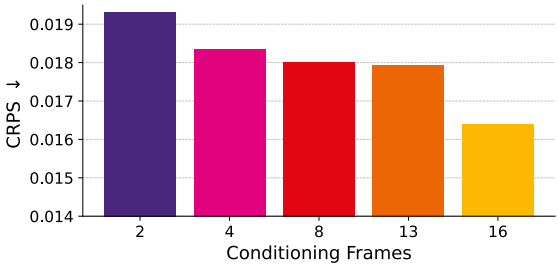


Figure S9. Performance of our forecasting pipeline for different numbers of conditioning frames.

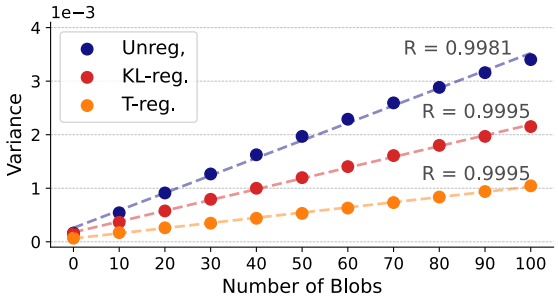
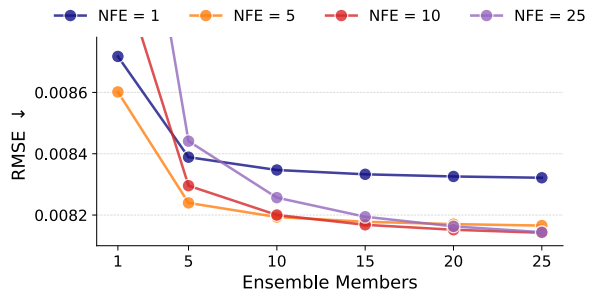


Figure S10. Intra-ensemble variance of first-stage reconstructions in the presence of increasing abnormal features (blobs). All linear regressions show significant correlations.

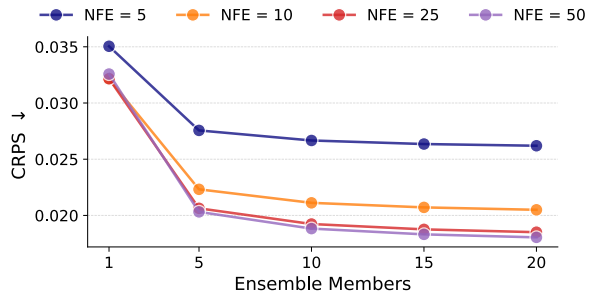
tect deviations from the training distribution.

We show qualitative samples of a reconstruction ensemble with blobs together with corresponding variance maps in Fig. S13 and Fig. S14. The variance maps display an outline of high variance surrounding blobs and lower variance within them. Therefore, variance is high where abnormal blobs interact with normal precipitation, as it is unclear how the precipitation will alter the blob shape. However, in the blob center, pixels are surrounded by extreme precipitation, leading to confidence in their extreme values. Due to the sharp outline, we can isolate abnormal patterns using variance maps. Therefore, the FREUD reconstruction ensembles can successfully detect and localize abnormal patches.

Scaling Test-time Compute Fig. S11a shows the influence of the ensemble size and the number of sampling steps for each ensemble member on the reconstruction performance of FREUD. We observe a clear benefit from larger ensemble sizes, which plateau after 10 ensemble members. The sampling steps do not show a consistent effect, and



(a) FREUD



(b) B-LSM

Figure S11. Ablation of scaling test time compute by increasing the number of function evaluations (NFE) and the ensemble size for the compression stage and B-LSM. Larger ensembles consistently improve the performance, and more function evaluations improve performance for the LSM.

many-step runs are outperformed by few-step sampling for small ensemble sizes. Due to the strong conditioning from the encoder, the generative decoding task is simple, and the increased generative capabilities of expensive sampling runs are outweighed by integration errors, explaining the observed effect. Therefore, we can use efficient few-step sampling, which limits the overhead from using a generative decoder. However, ensemble size and function evaluations seem to be entangled, as runs with many function evaluations benefit more from larger ensembles. We find a good trade-off with five function evaluations, which performs best up to 20 ensemble members.

Similarly, we evaluate the effect of ensemble size and sampling steps on the latent space model in Fig. S11b. CRPS consistently improves with a larger ensemble size and more function evaluations. Yet, the benefit of using more than 25 sampling steps and 15 ensemble members is marginal, revealing diminishing returns. Still, these results highlight that we can improve forecasting performance by scaling test time compute, enabling flexibility in deployment depending on the requirements for accuracy and latency.

Ensembling inference cost Our method outperforms CasCast with a single decoder ensemble (cf. Tab. 2), yet

Table 11. Inference time per decoder ensemble size on a single H200 GPU.

Ensemble size	1	5	10	15	20	25
Inference time (H200)	0.63s	2.49s	4.83s	7.18s	9.56s	11.93s

Table 12. Comparison of RaMViD to Diffusion Forcing training. RaMViD training yields better downstream forecasting skill across all metrics.

Training Paradigm	**CRPS**	**SSIM**	**HSS**	**CSI**
Diffusion Forcing	0.0218	0.7759	0.4627	0.3544
RaMViD	0.0196	0.7828	0.4923	0.3805

Fig. S11a shows larger ensembles improve performance. Since decoding is highly parallelizable, ensembling incurs limited wallclock overhead. Tab. 11 shows the evolution of wallclock latency with ensemble size. Even producing 25 decoder ensemble members requires less than 12 s inference time, remaining compatible with operational 5 min nowcasting.

Comparison with Diffusion Forcing Our masking-based training paradigm (RaMViD [54], see Sec. 3.3) can be interpreted as a special case of *Diffusion Forcing* [15]. In Diffusion Forcing, each frame is assigned an independent diffusion timestep during training. This offers more flexibility at inference time, as frames can be denoised using standard full-sequence, autoregressive, or rolling denoising. RaMViD can be interpreted as a variant of Diffusion Forcing, where during training, each frame is assigned either timestep $i = 1$ or timestep $i = \tau$, and during inference, each frame is initialized with a diffusion timestep $i = 1$ (data without noise) or $i = 0$ (pure noise).

We implement a variant of our model trained using Diffusion Forcing. Since our use case is forecasting, we further bias the sampling of per-frame timesteps so that temporally later frames tend to be assigned lower timesteps. Thus, on average, early frames are less noisy than later frames. This resembles the forecasting task at inference time. Tab. 12 compares RaMViD inference with the more flexible Diffusion Forcing approach using a B-LSM. We find superior performance with our masking-based training compared to the Diffusion Forcing setup across all metrics.

Table 13. Small (S), Base (B), and Large (L) latent space model (LSM) configurations.

Model	Layers	Hidden size	Attention Heads	Parameters
S-LSM	12	384	6	44M
B-LSM	12	768	12	141M
L-LSM	24	1024	16	473M

B. Implementation Details

In the following,€ we provide additional implementation details and hyperparameter settings.

B.1. Architecture

FREUD The FREUD encoder uses a 4×4 spatial patching followed by a downsampling layer, which is implemented as two transformer blocks with a hidden dimension of 96 and three attention heads, followed by a PixelShuffle [119] operation to halve the resolution. Thus, we achieve a total of $8\times$ downsampling along both spatial dimensions. The downsampling layers use neighborhood attention, where each pixel can only attend to a 7×7 spatial neighborhood. 2D Axial RoPE is used for positional embeddings. The encoder processes the downsampled inputs with four additional transformer blocks [28] with 384-dimensional tokens, six attention heads, and full self-attention to produce the final latent embeddings. For *T-reg.* FREUD, we additionally apply a Tanh function to the result and add a noise perturbation with $\sigma = 0.001$ to the latents. This small value is chosen to minimize adverse effects from perturbation and was not tuned or further ablated.

The FREUD decoder architecture is inspired by the Hourglass Diffusion Transformers (H-DiTs) [24], which enable efficient diffusion in pixel space with a transformer architecture. The FREUD decoder employs downsampling, similar to the encoder, with a $1 \times 4 \times 4$ patching and a downsampling layer. The decoder downsampling layer uses spatio-temporally factorized [10] neighborhood attention [45] with a 7×7 spatial and a 3-step temporal neighborhood. Therefore, each frame only attends to the immediate neighbor frames. In the bottleneck, we use 12 DiT blocks after concatenating the encoder latents channel-wise to the decoder feature maps. The upsampling layers of the decoder are built in parallel to the downsampling layers and use Token Split [119] operations for upsampling.

Latent Space Model The LSMs operate in the latent space of the FREUD first-stage. For improved convergence speed, we normalize latents to zero mean and unit variance with offsets determined on the training dataset. The LSMs use a standard DiT architecture [84, 99] with 2×2 patching. The specific settings for all model sizes are provided in Tab. 13. We use 3D Axial RoPE and factorized spatio-temporal attention for all variants.

B.2. Training

Multi-stage Training Training transformers on high-resolution spatio-temporal nowcasting data is expensive [35]. Exploiting RoPE's sequence length generalization capabilities [47, 132] and following current video training paradigms [135], we adopt a curriculum of training on progressively longer clips. We first pre-train FREUD and LSMs on single frames, and then progressively increase the number of frames per clip, until we finally train on the full sequence. During LSM image pre-training, we employ a standard unconditional diffusion training, which strengthens the model's spatial understanding before introducing the need to understand temporal dependencies. FREUD gains only slightly from full-sequence training, showing good length generalization, whereas the LSMs improve substantially when training on the full sequence. We train the FREUD encoder for 250k iterations and the LSM for 350k iterations.

Additional Hyperparameters We train with the AdamW [82] optimizer setting $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We use bfloat16 precision in training for efficiency. Image pretraining uses a 160 batch size on a single 40 GB A100. Later stages are run with a global batch size of 128. We use the Warmup-Stable-Decay (WSD) learning rate scheduler [55].

Outlier punishment Similar to MovieGen [135], we initially find spot artifacts in reconstructions in our FREUD first stage. Therefore, we adopt the outlier punishment from MovieGen and penalize deviations of latent values from the latent mean if they exceed r standard deviations. Following [135] we use

$$\mathcal{L}_{OPL}(\theta) = \frac{\sum_{i=1}^{H_l}\sum_{j=1}^{W_l} \max\left[||\mathbf{z}_{i,j} - \text{mean}(\mathbf{z})|| - r||\text{Std}(\mathbf{z})||, 0\right]}{H_l \cdot W_l}$$

where H_l and W_l are the dimensions of the latent embeddings, and $\mathbf{z} \in \mathbb{R}^{C_l \times H_l \times W_l}$ are the frame-wise latents. In practice, we set $\lambda_{OPL} = 10^5$ to a large value and use $r = 3$ as a common value in outlier detection. We observe $\mathcal{L}_{OPL}$ contributes only in the beginning of the training.

B.3. SEVIR dataset

The *SEVIR* dataset [139] is a weather observation dataset particularly well-suited for evaluating precipitation nowcasting methods because approximately 20% of the 20,393 weather events are taken from NOAA's Storm Event Database [94] and, hence, represent extreme weather. For the remaining 80% of the dataset, random events are sampled while giving a higher probability to high precipitation events. All events are drawn from the 2017–2019 time range and are recorded over the Continental United States.

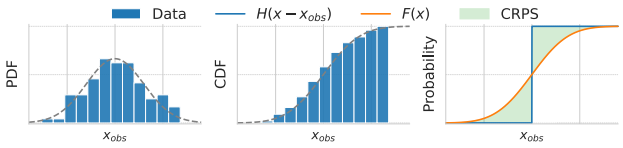


Figure S12. Schematic illustration of CRPS.

Each event covers a 384×384 km region over a 4 h timespan. To indicate precipitation, SEVIR uses the NEXRAD radar mosaic of Vertically Integrated Liquid (VIL). VIL is recorded with 1 km spatial and 5 min temporal resolution.

We use the same test data as Earthformer and CasCast [35, 40] to calculate our metrics and use the remaining data for training. Following previous work [35, 40, 115], we use 13 frames (65 min) as the conditioning to predict the next 12 frames (60 min) of precipitation unless specified otherwise. For the training data, we differ from previous work in taking exhaustive subsequences from each event. As recommended by the dataset creators [139], we keep the non-linear encoding of VIL from SEVIR and normalize the encoded $[0, 255]$ values to the $[-1, 1]$ range. At inference time, we can revert the normalization and apply the non-linear mapping

$$x_{kg/m^2} = \begin{cases} 0, & \text{if } x_{pixel} \leq 5 \\ (x_{pixel} - 2)/90.66, & \text{if } 5 < x_{pixel} \leq 18 \\ \exp((x_{pixel} - 83.9)/38.9) & 18 < x_{pixel} \end{cases}$$

to obtain the VIL in kg/m^2 [139].

For consistency with previous work [35, 40, 115] and alignment with the SEVIR data [139], we use $H = W = 384$, $C = 1$, $L^{in} = 13$, $L^{out} = 12$, and $T = 25$ unless specified otherwise.

B.4. Metrics details

To ensure comparability, we use the evaluation pipeline implemented by Gong et al. for the CasCast paper [40].

CRPS We calculate the CRPS to measure the alignment of the predicted distribution with the ground truth distribution. CRPS is a generalization of the Mean Absolute Error (MAE) to probabilistic predictions. CRPS is calculated as

$$\begin{aligned} CRPS(F, x) &= \int_{-\infty}^{\infty} (F(y) - \mathbf{1}_{y \geq x})^2 \, \mathrm{d}y \\ &= \mathbb{E}_{X \sim F} \left[||X - x|| \right] - \frac{1}{2} \mathbb{E}_{X, X' \sim F} \left[X - X' \right] \\ &\approx \frac{1}{N} \sum_{i=1}^{N} |f_i - x| - \frac{1}{2N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} |f_i - f_j|, \end{aligned}$$

where x is the observed value, $F(y)$ is the cumulative distribution function of the forecast, and $\{f_1, ..., f_N\}$ is the ensemble of forecasts. We show a schematic illustration of CRPS in Fig. S12. Essentially, CRPS is the squared error between the ground truth cumulative distribution and the predicted cumulative distribution, which can be approximated with a finite ensemble using the MAE of the forecast and subtracting the MAE of ensemble members.

SSIM Further, we use the Structural Similarity Index Measure (SSIM) to measure visual similarity between the ground truth precipitation and the forecast. SSIM for data with a single channel is calculated as

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)},$$

where x is the ground truth and y is the prediction, μ_x and μ_y are the average intensities, σ_x^2 and σ_y^2 are the variances of intensities, σ_{xy} is the covariance between the images and C_1 and C_2 are two small constants for numerical stability. SSIM is calculated by averaging results from a sliding window across the image and using Gaussian weighting to calculate means and variances, whereby the largest weight is assigned to the central pixel.

HSS and CSI We use the Heidke Skill Score (HSS) and Critical Success Index (CSI) computed on a per-pixel basis to identify positional inaccuracies. HSS is calculated as

$$HSS = \frac{2(TP \cdot TN - FP \cdot FN)}{(TP + FN)(FN + TN) + (TP + FP)(FP + TN)}.$$

Here, the True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) are calculated with respect to some threshold. To obtain the reported HSS, we average HSS values calculated using six thresholds (16, 74, 133, 160, 181, 219). HSS indicates improvement of a prediction over random chance, where 0 indicates no forecasting skill beyond chance and 1 indicates a perfect forecast.

CSI or Threat Score quantifies the proportion of correctly predicted events while excluding true negatives, which makes it useful in scenarios where non-events (e.g., no precipitation) are more common than true events. CSI is calculated as

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

Again, we report the CSI averaged over the six thresholds.

C. Limitations

Our method achieves superior distribution coverage and perceptual similarity as measured by CRPS and SSIM. However, especially the smaller models underperform for localization-centric metrics such as HSS and CSI, which aligns with previous work suggesting a trade-off between

localization and distribution coverage [153]. While our method achieves a superior trade-off compared to prior work, future work could explore introducing different conditionings and priors that improve localization while keeping full distribution coverage. A possible solution might be to use noisy prior frames as the starting point for flow matching [76], which, however, is non-trivial with non-autoregressive video modeling, as in RaMViD.

Our method yields more accurate and better-calibrated forecasts than the current state-of-the-art. Furthermore, FREUD can quantify reconstruction uncertainty at inference time, which was not possible with previous VAE-based compression stages. Sampling from the FREUD decoder achieves better calibration than samples from a VAE. Still, our forecasts and FREUD reconstructions exhibit some overconfidence. Therefore, future work should focus on developing calibration methods for generative flow models. Related to this problem, our method, as well as prior methods, tend to underestimate precipitation. As generative models learn the distribution of the training data, this is expected because low precipitation is more common than extreme precipitation. Therefore, future work should explore options to better align the forecast distribution with the ground truth in rare extreme scenarios.

Previous work has solved this problem using cfg [48]; however, we find that the use of cfg is flawed in precipitation nowcasting, as increasing cfg leads to higher precipitation independent of the conditioning. This holds for our method as well as our strongest competitor, CasCast (see Sec. A.4). Therefore, the use of cfg does not yield improved alignment with conditions, but rather increases intensity. Follow-up research should identify better guidance mechanisms that enable the generation of realistic forecasts that align with actual conditions.

As with all generative models, our approach inherits the statistical properties of its training data. Since high-quality weather datasets are primarily collected in technologically advanced regions with dense radar coverage, nowcasting methods for precipitation that rely on radar observations - including ours - are inherently limited to areas with such infrastructure, leaving large parts of the world (e.g., oceans and many developing regions) underserved. Future work should explore conditioning strategies that do not depend solely on radar. Cloud-top temperatures from geostationary satellites offer near-global coverage [134], and decades of research have demonstrated their utility for estimating precipitation in radar-sparse regions [52, 60, 87, 113]. These alternative conditioning signals present a promising path toward globally applicable precipitation nowcasting.

D. Weather Forecasting Literature Review

Weather nowcasting refers to short-term weather forecasting for periods of 30 min to 12 h, with high temporal and spatial resolution [31, 37, 40, 71]. Traditionally, numerical weather prediction (NWP) systems, which simulate atmospheric evolution by numerically solving physical equations, have set the state-of-the-art for this task [29, 31, 108, 125]. NWPs approximate solutions by averaging over spatial regions. For low-latency forecasts, higher downsampling is required, resulting in reduced resolution [74]. Further, the inherent non-linearity in atmospheric dynamics [69, 91, 131] implies that minor perturbations or sensor inaccuracies will lead to exponential divergence over time [64, 81, 96]. Therefore, NWPs are run multiple times with slightly altered inputs to estimate uncertainty, exploding computational costs. As a computationally efficient alternative to NWPs, extrapolation-based techniques using optical flow are used [22, 39, 114, 146, 147]. While these methods are much faster to compute, they are flawed because optical flow cannot model the formation and dissipation of weather patterns, limiting their predictive capability.

To overcome the limitations of traditional methods, recent work has turned to deep learning for efficient precipitation nowcasting. These approaches typically fall into two categories: *deterministic* and *probabilistic* models. Deterministic nowcasting refers to models that make a single prediction based on the input and do not account for uncertainty in their predictions. In contrast, probabilistic nowcasting methods usually employ generative models to produce samples of possible outcomes, which allows for estimating uncertainty through repeated sampling.

D.1. Deterministic

Recurrent In an early work, Shi et al. [120] propose the ConvLSTM architecture, which extends the traditional LSTM [51] with convolutions to retain spatial structure. With this modification, they outperform a traditional optical flow-based nowcasting system [146]. Later, they proposed the TrajGRU [121], which improves the previous method by replacing fixed-size convolution with learned warping. The PredRNN [144] extends ConvLSTMs with a dual memory mechanism, integrating short-term cell and long-term memory, to better capture complex spatio-temporal dependencies. PhyDNet [42] outperforms the previous methods by learning a latent space with known dynamics, evolving the latent state according to these laws, and using a ConvLSTM to correct inaccuracies.

Convolutional Differing from the previous methods, Agrawal et al. [1] formulate nowcasting as an image-to-image translation problem and use a convolutional Unet architecture [111] with timesteps concatenated along the channel dimension. Their method outperforms optical flow and a low-latency NWP [29]. Trebing et al. [136] improve computational efficiency by factorizing spatio-temporal attention and convolutions, reducing the number of parameters while retaining competitive performance. In contrast, Fernàndez & Mehrkanoon [33] explicitly model the temporal dimension using 3D convolutions instead of channel-wise concatenation and retain computational efficiency by factorizing 3D convolution into three axial convolutions. Additionally, convolutions with increasing dilation are used in the bottleneck. Their method outperforms the previous convolutional approaches, highlighting the advantage of temporal modeling. Gao et al. propose "simpler yet better video prediction" (SimVP) [36], which uses a frame-wise encoder and decoder and a convolution-based bottleneck model that captures temporal evolution. Their approach outperforms recurrent models [42, 120, 144], underscoring the advantage of treating nowcasting as image translation.

Transformers Yang et al. [151] propose a TransUnet [17] architecture for nowcasting where transformer blocks [28, 138] are used in a Unet bottleneck. They further augment the Unet with attention and apply factorized convolutions, resulting in improved performance over a standard TransUnet and previous convolutional Unet architectures, revealing benefits of transformer-based modeling. Discarding convolution entirely, Gao et al. [35] introduce the Earthformer model with cuboid attention for efficient processing of high-resolution spatio-temporal data. Cuboid attention is applied to fixed-size spatio-temporal regions independently, and communication between cuboids is facilitated by global self-attention over class tokens. Their approach outperforms convolutional [1] and recurrent [42, 120, 144] architectures, cementing the advantage of attention-based modeling. Similarly, Pathak et al. [97] also adopt a ViT-based [28] architecture but compute self-attention in the Fourier domain for efficiency. They outperform a state-of-the-art NWP [109] for small-scale variables such as precipitation, demonstrating the superiority of deep learning for nowcasting.

Classification In contrast to the previous methods, Sonderby et al. [125] treat nowcasting as classification and predict a SoftMax probability distribution over discrete intensity bins. Their approach, MetNet, conditions on a large spatial context of previous precipitation, cloud-top temperatures, and topology and uses a spatial encoder for efficient processing with a bottleneck ConvLSTM and Axial Self-Attention [49]. MetNet-2 [31] uses an even larger context, and an additional sequence of convolutions with increasing dilation to outperform a state-of-the-art NWP ensemble [89] over the entire lead time range up to 12 h, demonstrating the flexibility of deep learning-based nowcasting.

The major drawback of these deterministic methods is their lack of accurate uncertainty quantification. Methods trained with regression losses minimize the mean difference

between predicted and observed values, where the minimizer is given as the expectation over the outcome, leading to mode averaging, which results in blurry predictions. Models trained with a classification objective mitigate this problem, but the pixel-wise objective does not exploit spatial dependencies and cross-correlations. In addition, SoftMax probability distributions suffer from suboptimal calibration [57].

D.2. Probabilistic

GAN nowcasting As a solution, Ravuri et al. [107] propose to use a conditional Generative Adversarial Network (GAN) [41, 88] with a temporal and spatial discriminator for nowcasting, where samples from the model are sharp and multiple samples can be produced to quantify uncertainty. Compared to deterministic methods such as MetNet [125] and Unet [1], and extrapolation (exemplified by PySteps [59, 104]), their method, Deep Generative Models of Radar (DGMR), achieves superior accuracy, calibration, and expert evaluation. Ji et al. [61] propose a ConvLSTM-based GAN model that outperforms optical flow and ConvLSTM for heavy precipitation events, underscoring the importance of probabilistic modeling in critical extreme situations. Liu et al. [79] propose GAN-based forecasting with conditioning on a deterministic forecast, resulting in sharpened forecasts compared to the deterministic-only approach. A similar approach is proposed by Price & Rasp [102] who condition a GAN on a low-resolution NWP ensemble forecast to produce a high-resolution prediction. Their method is superior to simple interpolation and convolution-based upsampling and approaches the performance of a state-of-the-art NWP [89] with drastically reduced complexity. Finally, Zhang et al. [158] propose NowCastNet, which combines a differentiable extrapolation with a learned residual and a stochastic GAN refiner. Their approach outperforms previous deterministic [144], probabilistic [107], and extrapolation-based [59, 104] methods according to quantitative metrics and expert evaluations, highlighting the importance of sharp generative forecasts for downstream deployment.

While these works hint at the importance of probabilistic modeling for nowcasting, they rely on GANs, which are known to suffer from mode collapse [66]. Therefore, the samples from these models will underestimate the true weather variance. Further, they might neglect additional smaller modes in the true weather distribution, which are particularly important in nowcasting rare extreme weather events. Therefore, these models fail to capture the full distribution of future weather evolution.

Diffusion-based nowcasting To tackle this problem, Leinonen et al. [71] use a diffusion model with a solid mathematical foundation [50, 124] and empirically high sample variance [50, 93] for nowcasting. Due to the complexity of iterative sampling, they opt for a latent diffusion approach [110], and apply diffusion in the latent space of a VAE compression model. Their generative model is conditioned on a deterministic prediction, which is augmented by the diffusion model. This architecture outperforms GAN-based nowcasting [107], especially for detecting extreme weather events. Similarly, Gao et al. [37] use a latent diffusion approach for their PreDiff architecture, but discard the deterministic conditioning in favor of physics-based guidance and conditioning on previous precipitation maps. Their approach consistently outperforms a range of deterministic [1, 35, 42, 120, 144] and GAN-based [107] approaches.

Since these early successes, architectural modifications have been proposed. Asperti et al. [7] suggest a learned aggregation of ensemble members instead of the canonical mean, while She et al. [115] integrate a transformer block from a large vision language model to ease training. Nai et al. [90] identify classifier-free guidance [48] as an option to improve forecasting skill. Moreover, Ling et al. [77] raise concerns about latent diffusion, as the cascaded architecture can lead to error accumulation and unquantified uncertainty. As an alternative, they suggest training the model end-to-end while integrating conditioning by concatenating feature maps from an condition encoder model.

Diffusion has also been used for longer-range forecasts. Li et al. [74] forecast precipitation over 16 days with a latent diffusion model conditioned on a deterministic forecast and outperform two NWP approaches [100, 162], cementing diffusion-based weather forecasting as a viable alternative to simulation. Further, Stock et al. [130] show that by auto-regressively unrolling a 1 h ahead diffusion-based forecast, reasonable forecasts up to 60 days lead time can be obtained and seasonal patterns are captured in year-long rollouts, indicating a robust understanding of the weather system in diffusion models. Similarly, Shi et al. [118] outperform a range of deterministic baselines [19, 92, 97, 140] with auto-regressive modeling conditioned on only two initial states. Building on this success, Price et al. [103] introduce GenCast as a follow-up to GraphCast [68], providing 15-day global forecasts of 25 variables with 27 km spatial and 12 h temporal resolution. The model is conditioned on two previous states, forecasts auto-regressively, and uses a Graph Transformer [156] backbone. They outperform a top-of-the-line ensemble NWP system [89], demonstrating the superiority of deep-learning-based generative forecasting with diffusion. Recently, Alet et al. [5] proposed *Functional Generative Networks* (FGN) as a follow-up to GenCast. They independently train an ensemble of predictors to account for epistemic uncertainty. To model aleatoric uncertainty, the authors turn the ensemble members themselves probabilistic by sampling a low-dimensional noise vector and injecting it via conditional normalization lay-

ers, effectively acting as a structured weight perturbation. The models are then trained directly using CRPS as the loss function, encouraging the models to make accurate and diverse predictions. The proposed model outperforms GenCast while providing substantially faster generation. Their results call the current dominance of diffusion-based approaches into question, although it remains unclear how well this approach transfers to domains where fine-scale stochasticity plays a larger role, such as precipitation nowcasting.

D.3. Cascaded deterministic-probabilistic

Early results show that conditioning a diffusion model on a deterministic forecast yields high-quality forecasts [18, 71, 72, 74, 77, 118]. Yu et al. [153] analyze these results and determine that probabilistic diffusion-only approaches show superior distribution coverage and sharpness than deterministic forecasts, but often suffer in terms of accurate localization. Building on physical knowledge, they suggest using a diffusion model to predict the residual between a deterministic forecast and the ground truth and find improved performance over deterministic-only [35, 36] or diffusion-only [37] approaches. Similarly, CasCast [40] conditions a latent space diffusion transformer [99] on a deterministic forecast and applies sequence-wise diffusion transformer blocks after spatial encoding. Their model consistently outperforms deterministic [35–37, 42, 120], GAN-based [157], and diffusion-only [37] nowcasting methods. Inspired by these results, Pathak et al. [98] condition an auto-regressive diffusion model on a deterministic initial forecast, and find their model outperforms a low-latency NWP [29]. However, they also indicate their method is overconfident and not well calibrated, hinting at a disadvantage of cascaded architectures.

We improve upon these prior works by compressing pixel-space data into a low-resolution latent space while quantifying the uncertainty from compression [77]. Further, we avoid biasing the generation with a deterministic forecast [98] and aim to achieve similar localization by exploiting the advantages of transformer-based architectures that allow to exploit conditioning more effectively with self-attention.

E. Uncurated Qualitative Results

In the following we provide further qualitative samples.

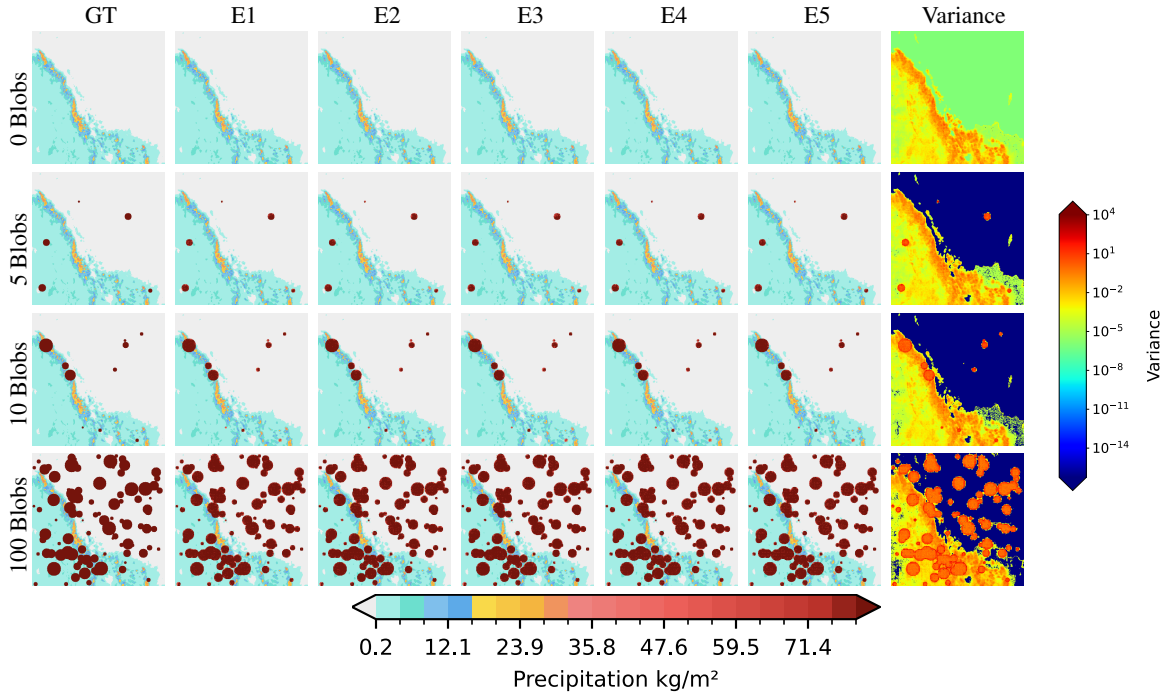


Figure S13. Qualitative results for blob experiment. The variance around blobs is high. Therefore, our first stage can localize abnormal features using ensemble variance. Best viewed zoomed in.

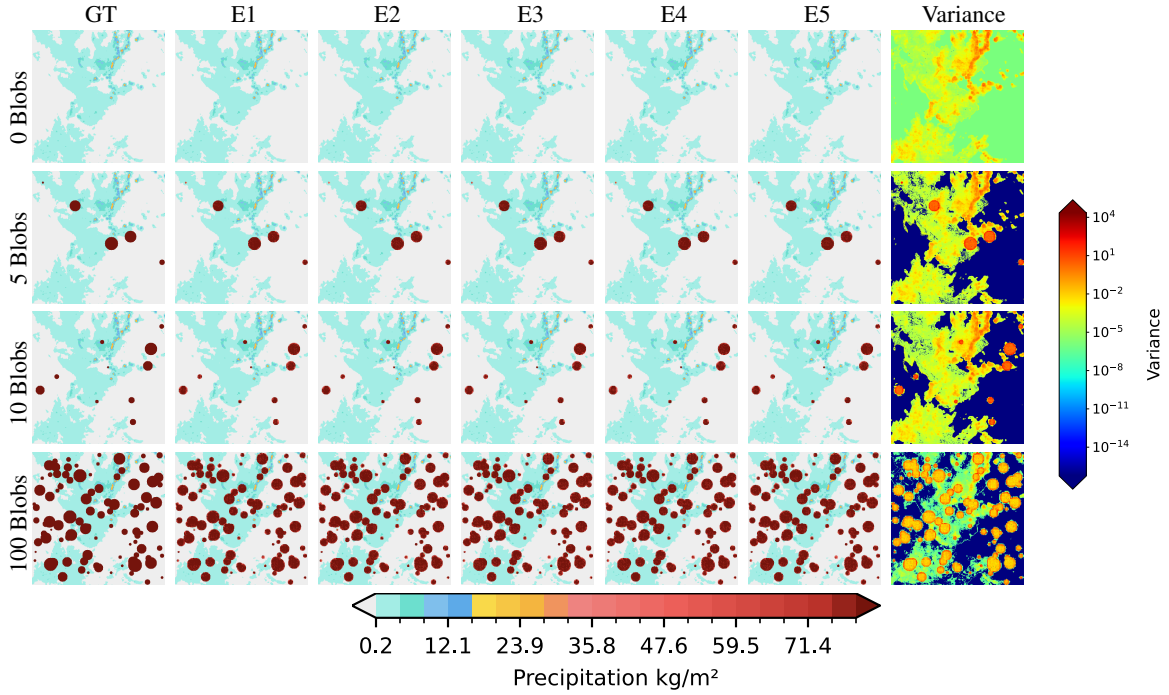


Figure S14. Qualitative results for blob experiment. The variance around blobs is high. Therefore, our first stage can localize abnormal features using ensemble variance. Best viewed zoomed in.

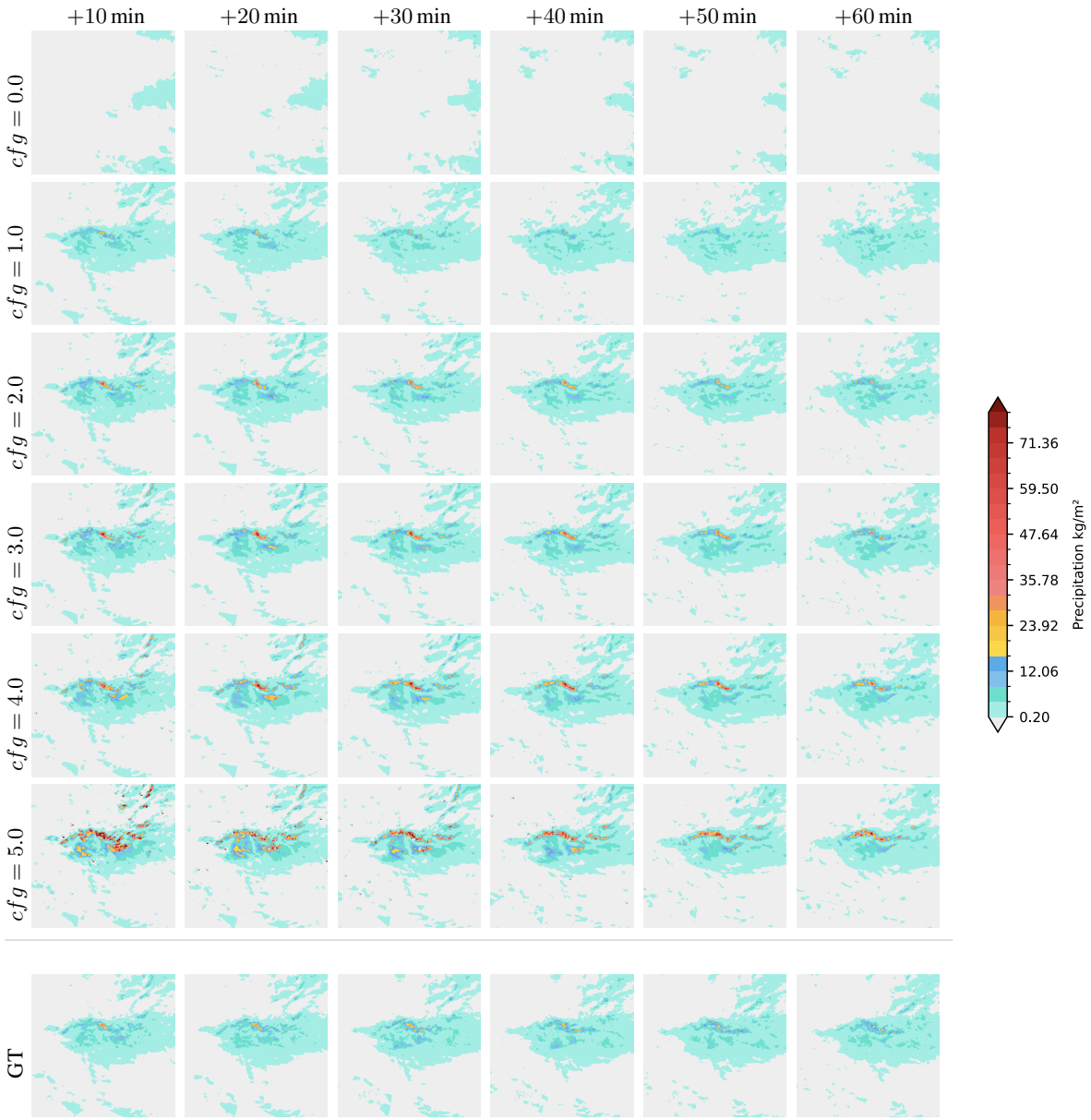


Figure S15. Qualitative results obtained with our B-LSM in the *T-reg.* latent space for different guidance scales [48]. Guidance 0.0 indicates unconditional sampling, while guidance 1.0 indicates conditional sampling without guidance. Best viewed zoomed in.

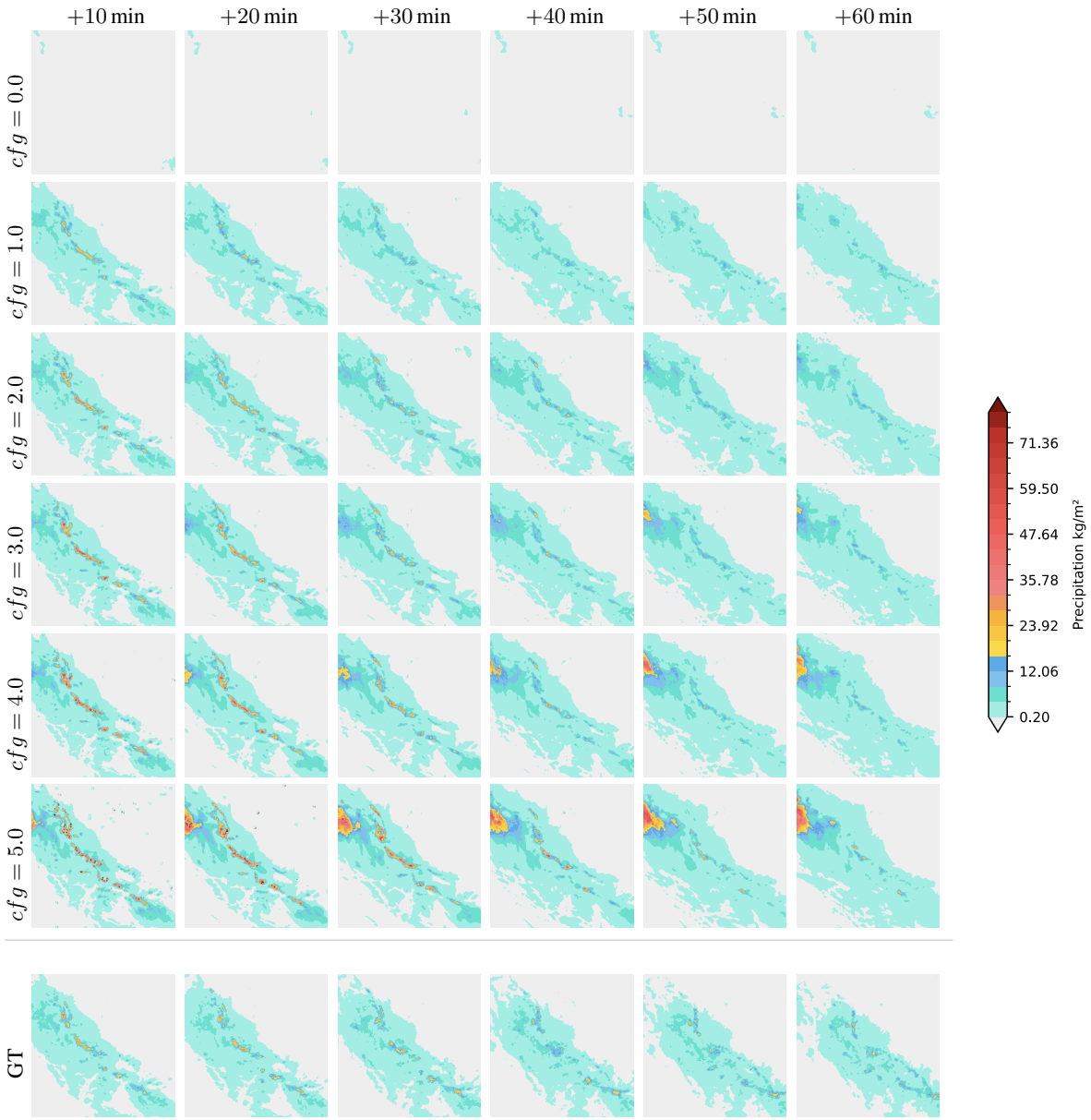


Figure S16. Qualitative results obtained with our B-LSM in the *T-reg.* latent space for different guidance scales [48]. Guidance 0.0 indicates unconditional sampling, while guidance 1.0 indicates conditional sampling without guidance. Best viewed zoomed in.

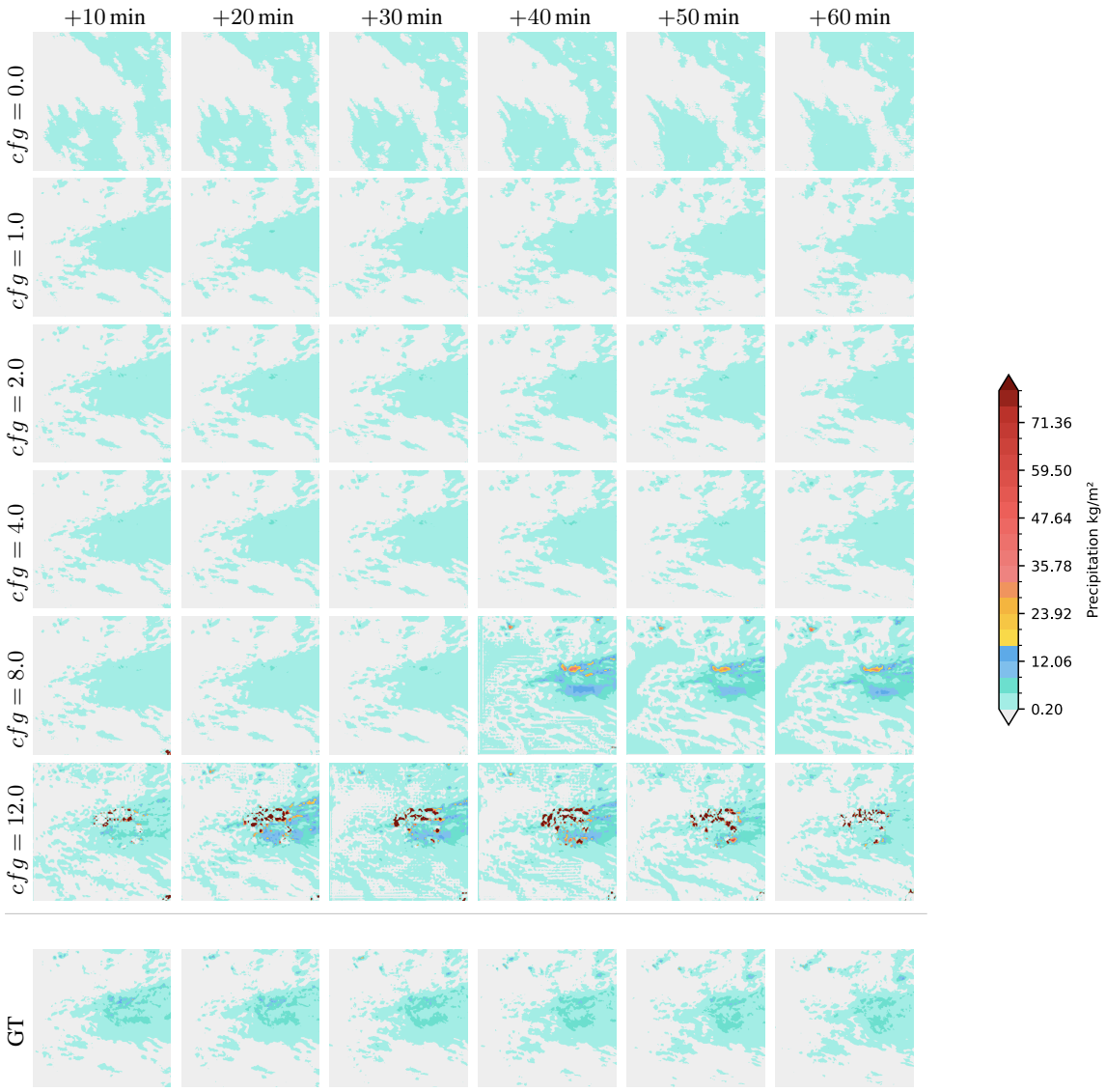


Figure S17. Qualitative results obtained with **CasCast** [40] for different guidance scales [48]. Guidance 0.0 indicates unconditional sampling, while guidance 1.0 indicates conditional sampling without guidance. Best viewed zoomed in.

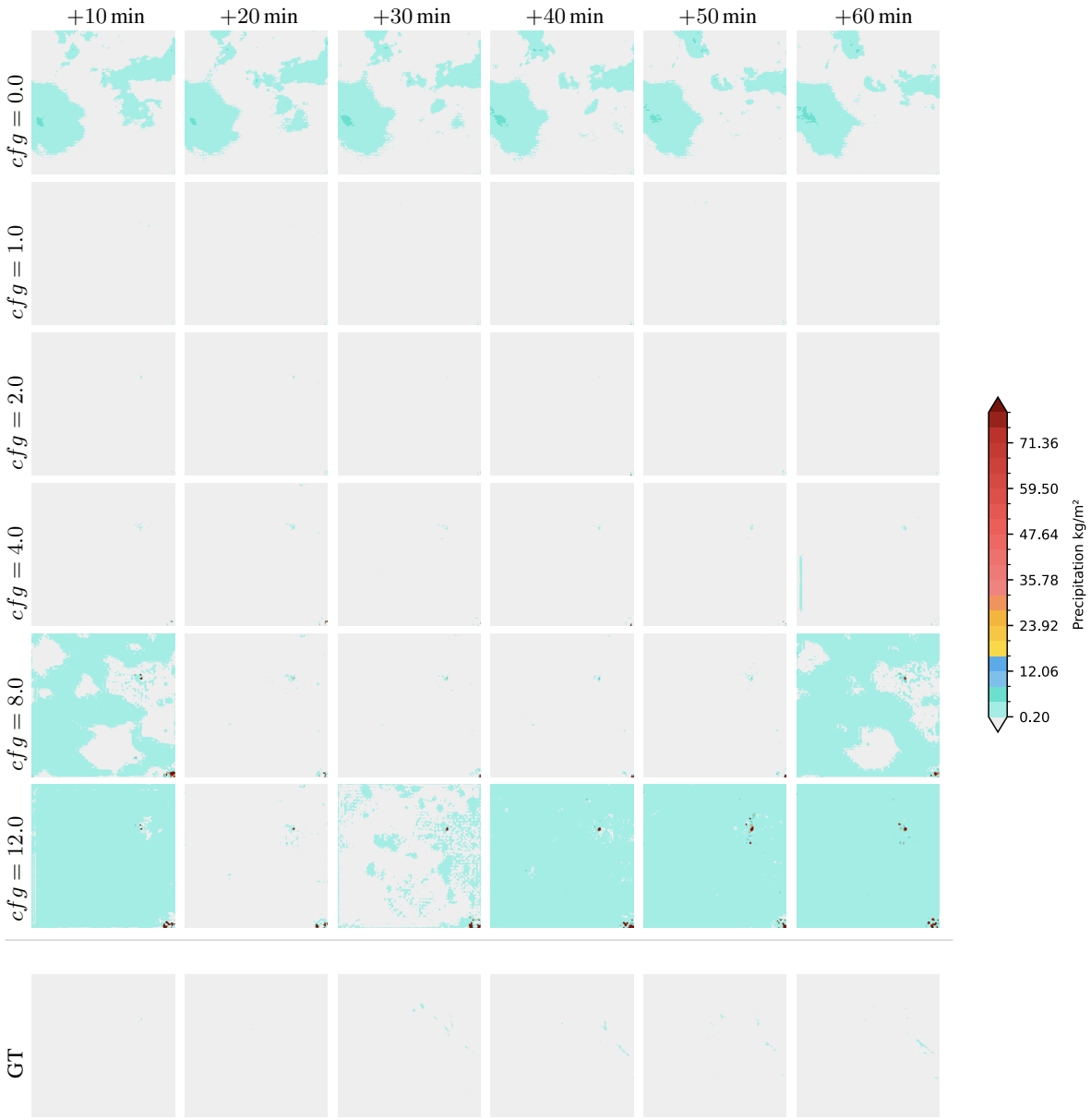


Figure S18. Qualitative results obtained with **CasCast** [40] for different guidance scales [48]. Guidance 0.0 indicates unconditional sampling, while guidance 1.0 indicates conditional sampling without guidance. Best viewed zoomed in.

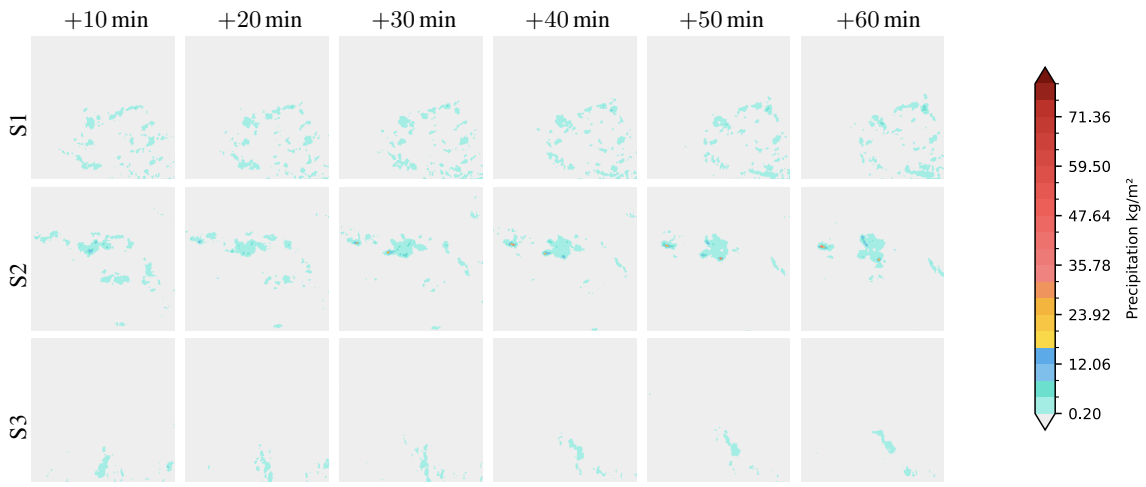


Figure S19. *Unconditional* qualitative samples from the B-LSM in *T-reg.* latent space. We observe that unconditional samples are temporally consistent, and sharp, but tend to contain low precipitation.

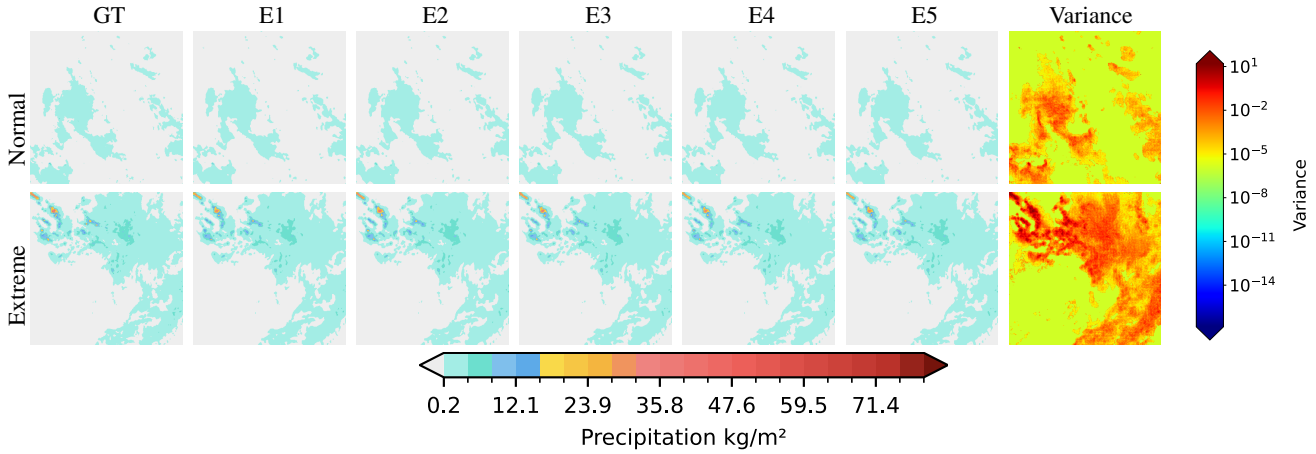


Figure S20. Qualitative reconstruction sampled with the frame-wise DiffAE. Qualitatively, the frame-wise reconstructions match the ground truth well, validating the strong reconstruction performance of the generative decoder.

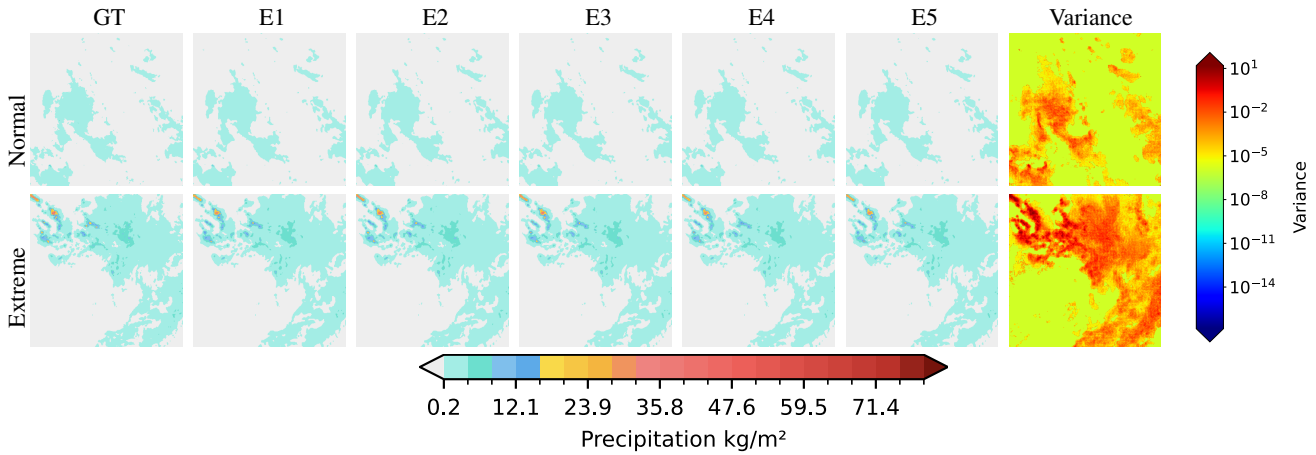


Figure S21. Qualitative reconstruction sampled with the frame-wise DiffAE. Qualitatively, the frame-wise reconstructions match the ground truth well, validating the strong reconstruction performance of the generative decoder.

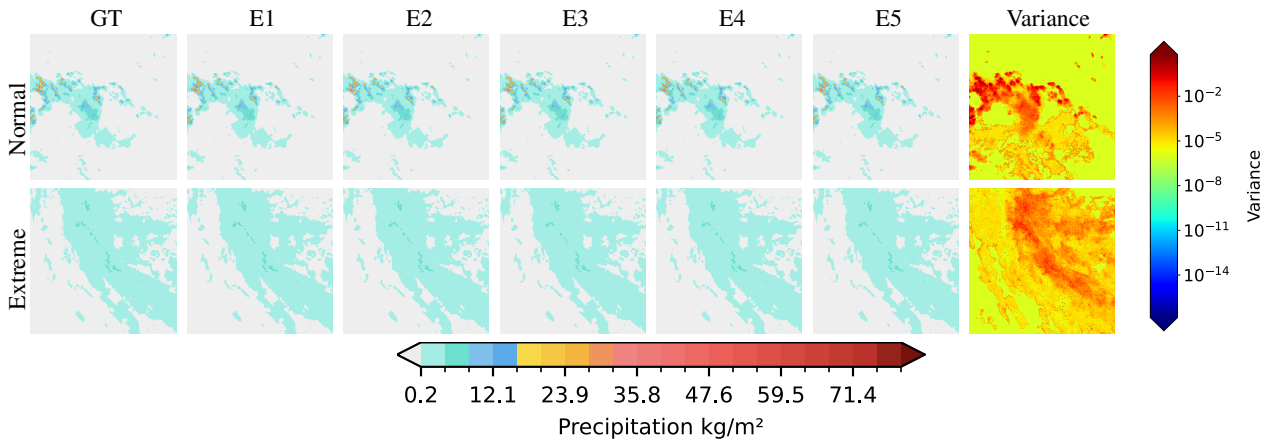


Figure S22. Qualitative results for reconstruction of one normal and one extreme weather event. Variance map uses log scale. Areas with precipitation show higher variance. Areas with extreme precipitation exhibit extreme variance. Best viewed zoomed in.

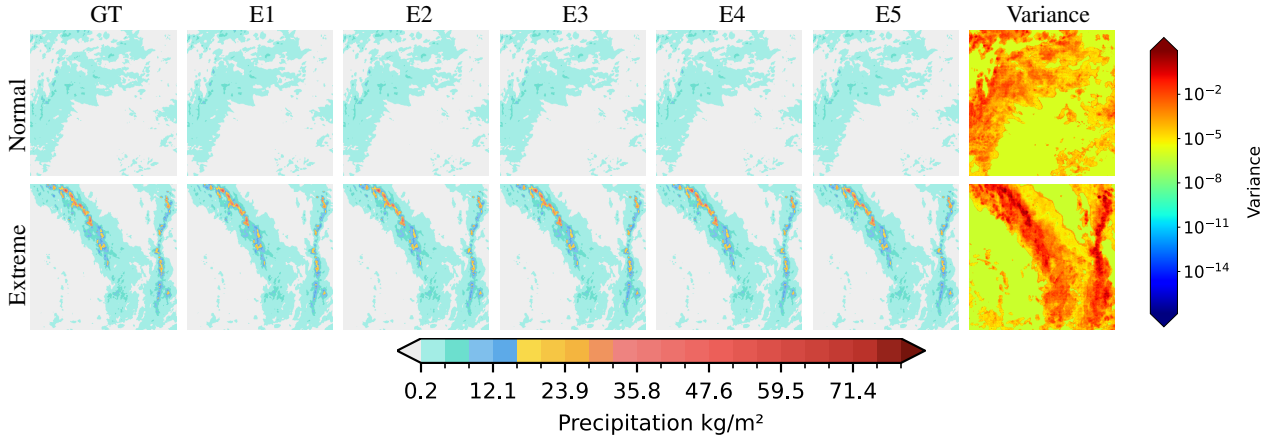


Figure S23. Qualitative results for reconstruction of one normal and one extreme weather event. Variance map uses log scale. Areas with precipitation show higher variance. Areas with extreme precipitation exhibit extreme variance. Best viewed zoomed in.

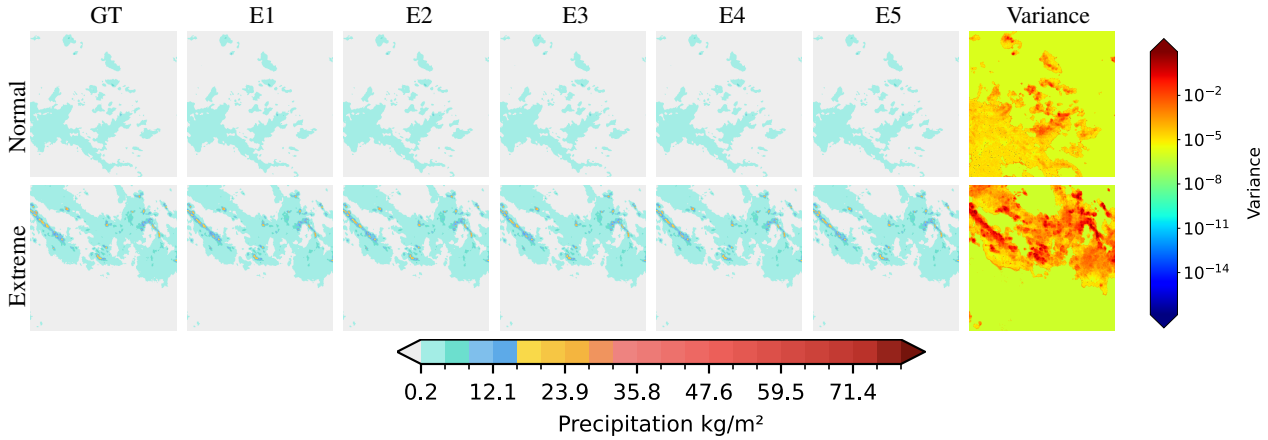


Figure S24. Qualitative results for reconstruction of one normal and one extreme weather event. Variance map uses log scale. Areas with precipitation show higher variance. Areas with extreme precipitation exhibit extreme variance. Best viewed zoomed in.

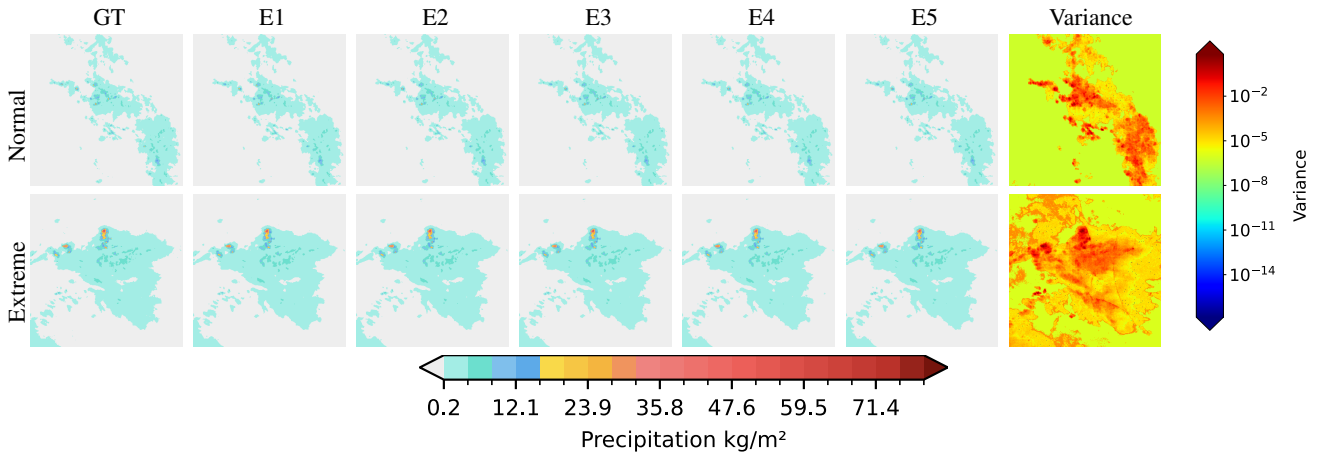


Figure S25. Qualitative results for reconstruction of one normal and one extreme weather event. Variance map uses log scale. Areas with precipitation show higher variance. Areas with extreme precipitation exhibit extreme variance. Best viewed zoomed in.

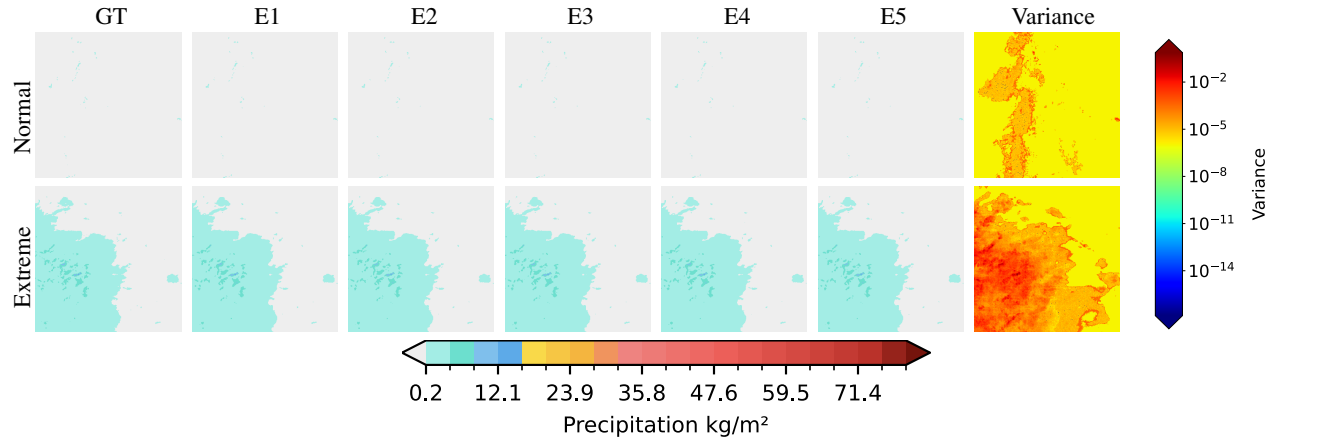


Figure S26. Qualitative results for reconstruction of one normal and one extreme weather event. Variance map uses log scale. Areas with precipitation show higher variance. Areas with extreme precipitation exhibit extreme variance. Best viewed zoomed in.

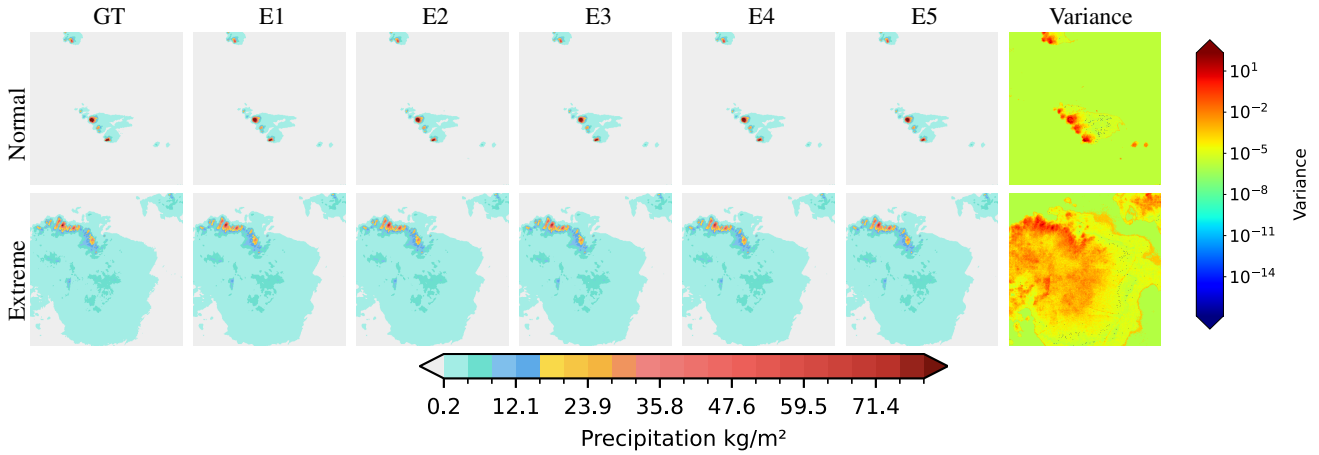


Figure S27. Qualitative results for reconstruction of one normal and one extreme weather event. Variance map uses log scale. Areas with precipitation show higher variance. Areas with extreme precipitation exhibit extreme variance. Best viewed zoomed in.

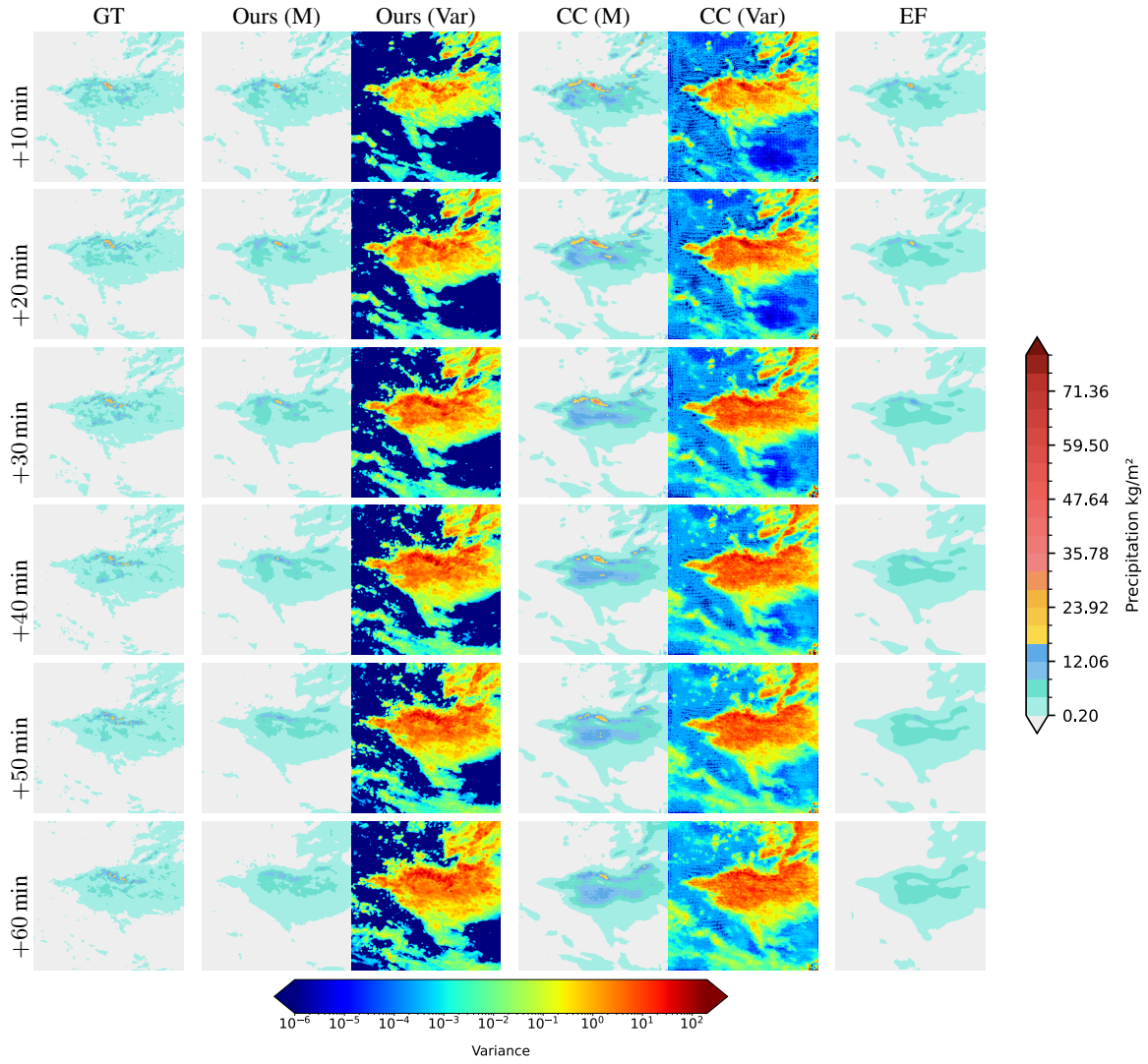


Figure S28. Comparison of forecasts made with our method, CasCast (CC) [40] and Earthformer (EF) [35]. For probabilistic methods, we show the mean (M) forecast and variance (Var) of a 10-member ensemble. Our forecast aligns more closely with the ground truth and variance, while CasCast overestimates and Earthformer underestimates precipitation. Further, the ensemble variance in our method focuses on high precipitation regions in the ground truth. Best viewed zoomed in.

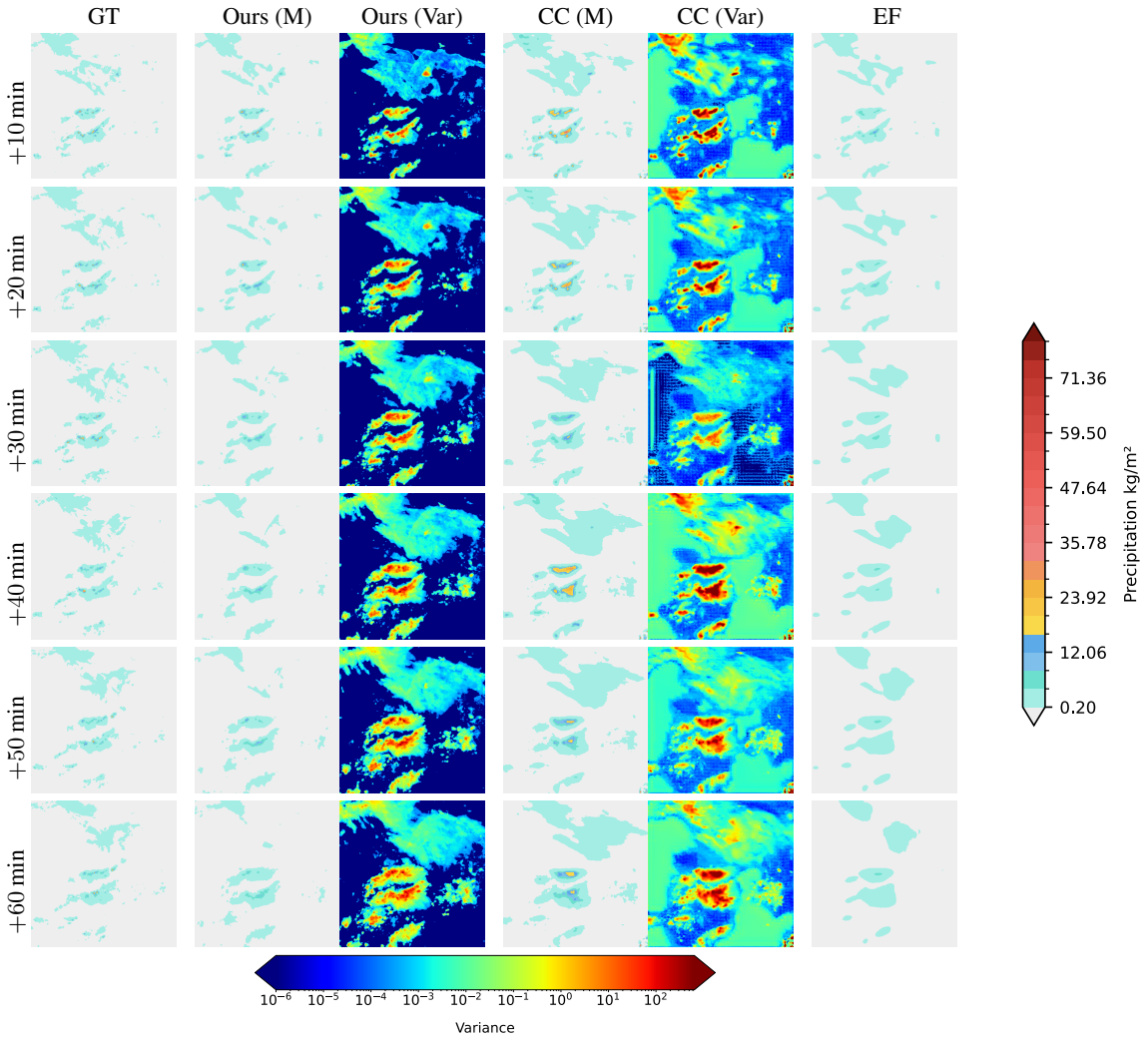


Figure S29. Comparison of forecasts made with our method, CasCast (CC) [40] and Earthformer (EF) [35]. For probabilistic methods, we show the mean (M) forecast and variance (Var) of a 10-member ensemble. Our forecast shows substantially improved temporal consistency compared to CasCast, where a strong precipitation event appears for a single frame. Further, the shape of our forecast aligns more closely with the ground truth. Best viewed zoomed in.

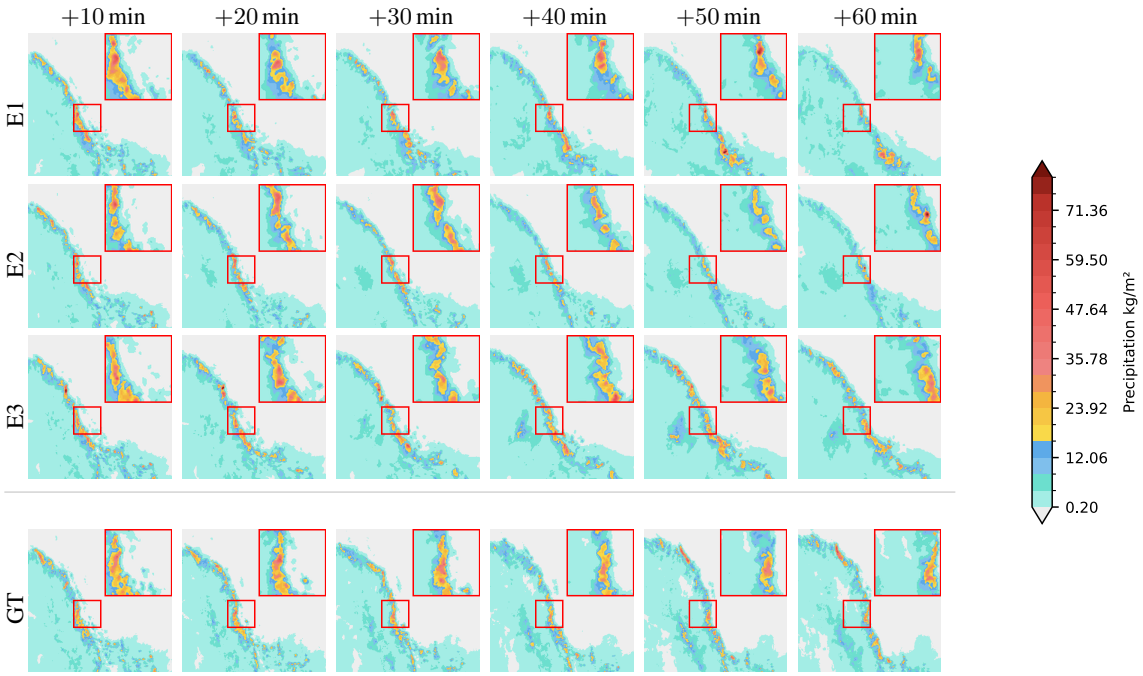


Figure S30. Qualitative result for our L-LSM with *T-reg.* first-stage. The red rectangle highlights a punch-in for better visualization of ensemble differences in details. Best viewed zoomed in.

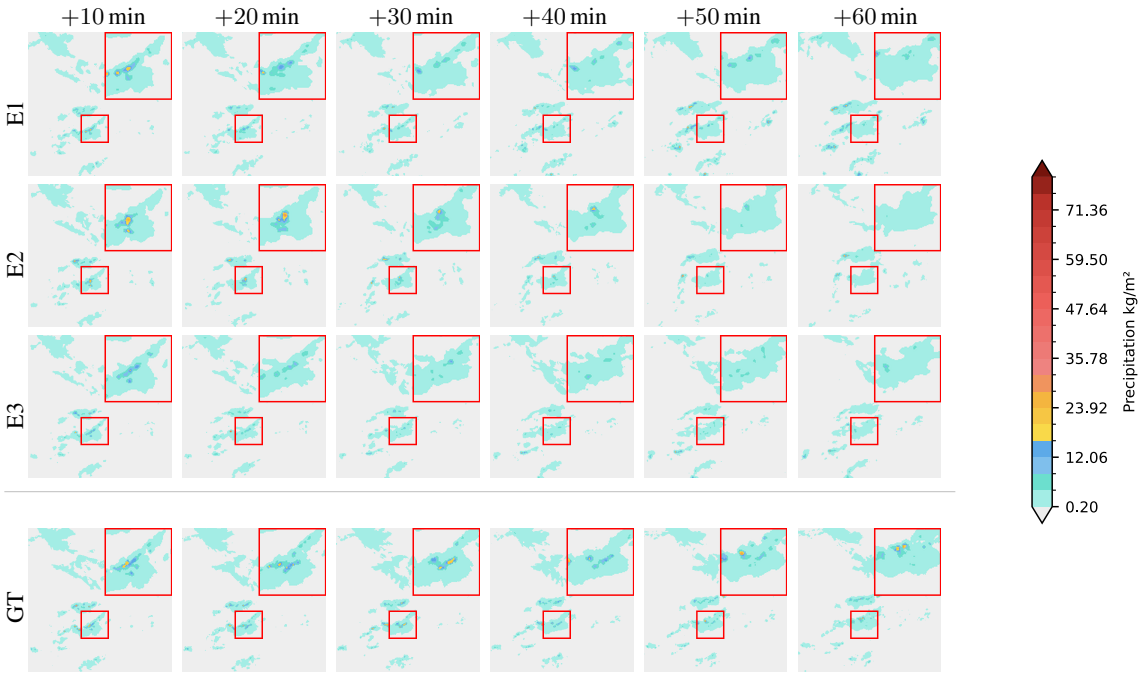


Figure S31. Qualitative result for our L-LSM with *T-reg.* first-stage. The red rectangle highlights a punch-in for better visualization of ensemble differences in details. Best viewed zoomed in.

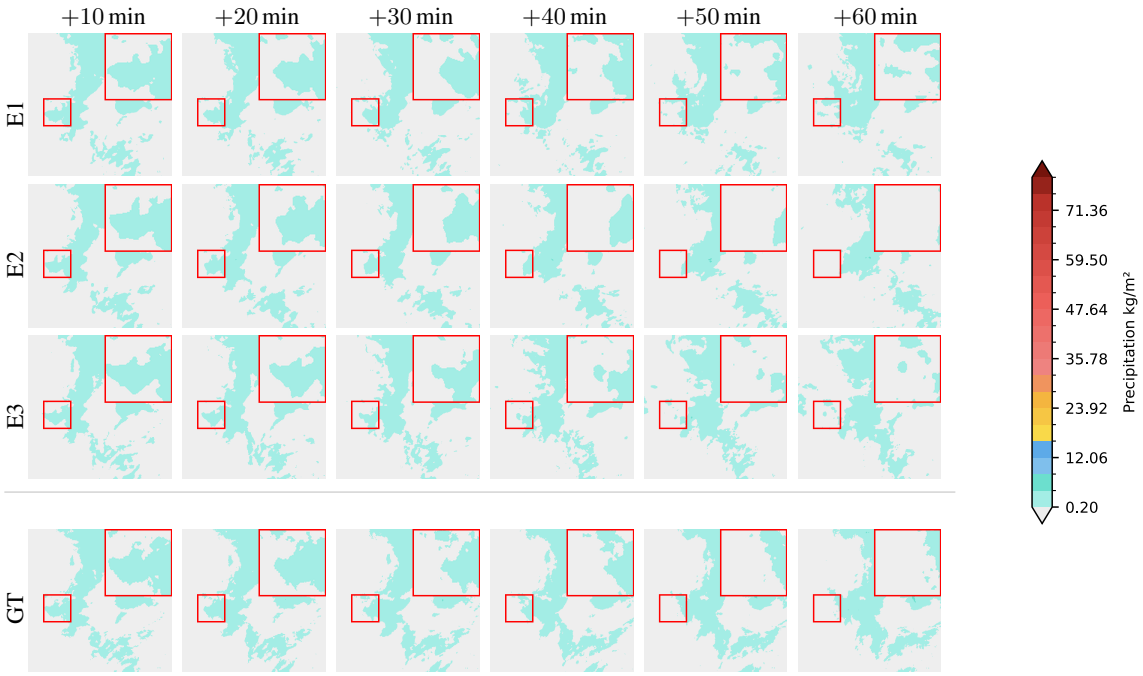


Figure S32. Qualitative result for our L-LSM with *T-reg.* first-stage. The red rectangle highlights a punch-in for better visualization of ensemble differences in details. Best viewed zoomed in.

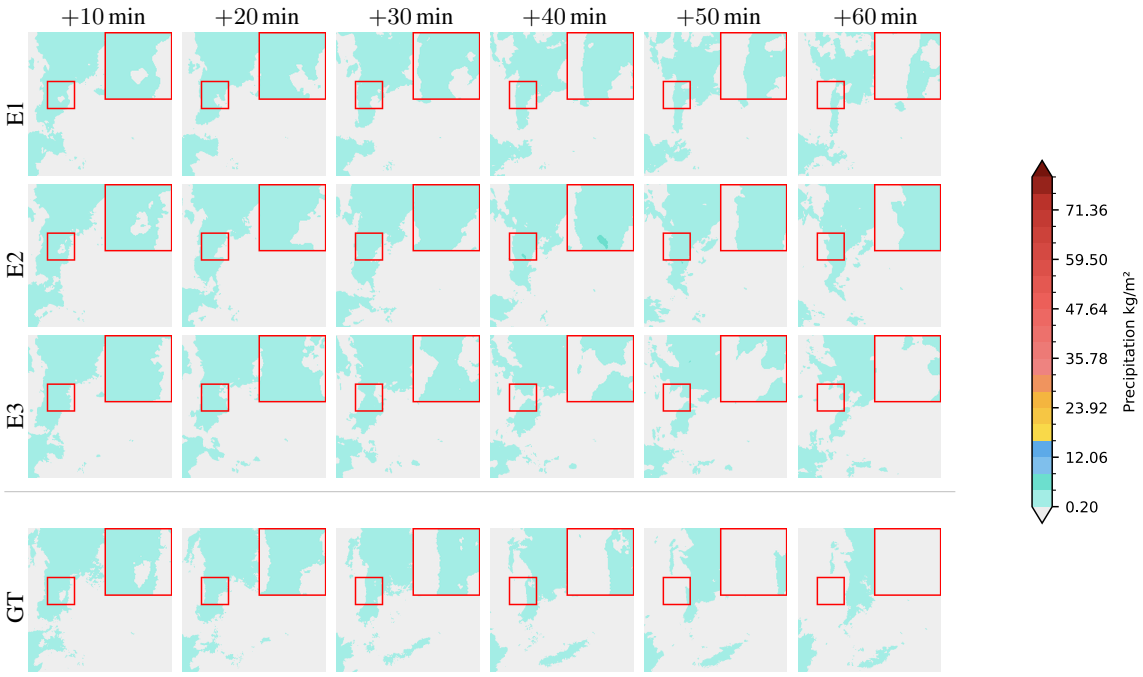


Figure S33. Qualitative result for our L-LSM with *T-reg.* first-stage. The red rectangle highlights a punch-in for better visualization of ensemble differences in details. Best viewed zoomed in.

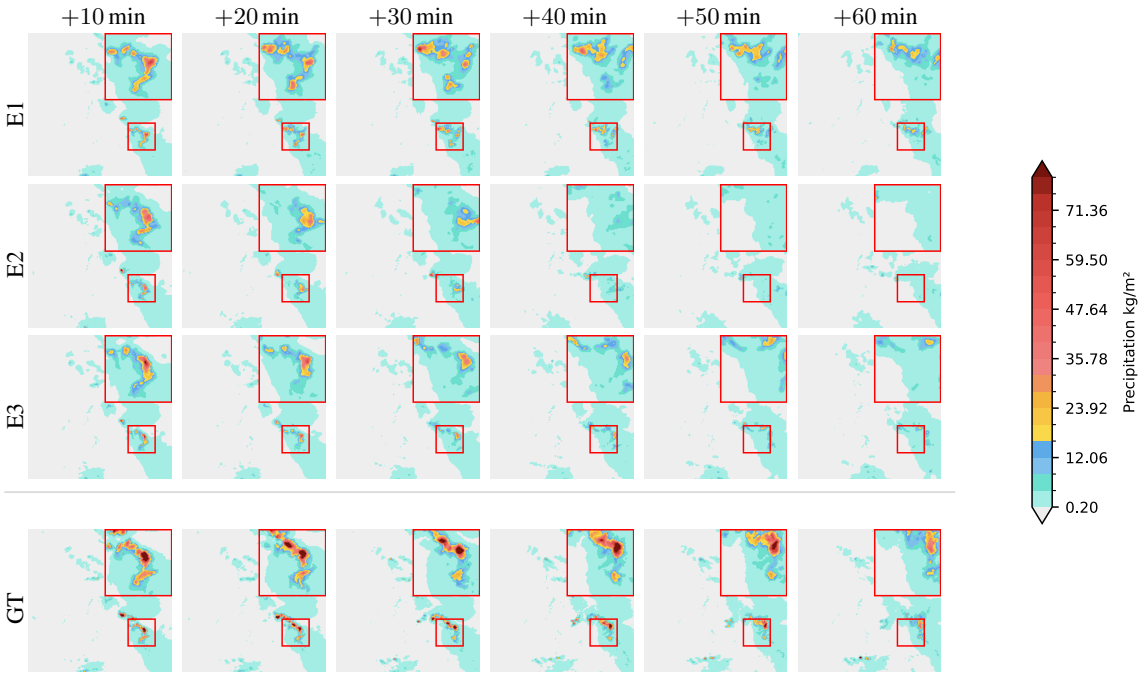


Figure S34. Qualitative result for our L-LSM with *T-reg.* first-stage. The red rectangle highlights a punch-in for better visualization of ensemble differences in details. Best viewed zoomed in.

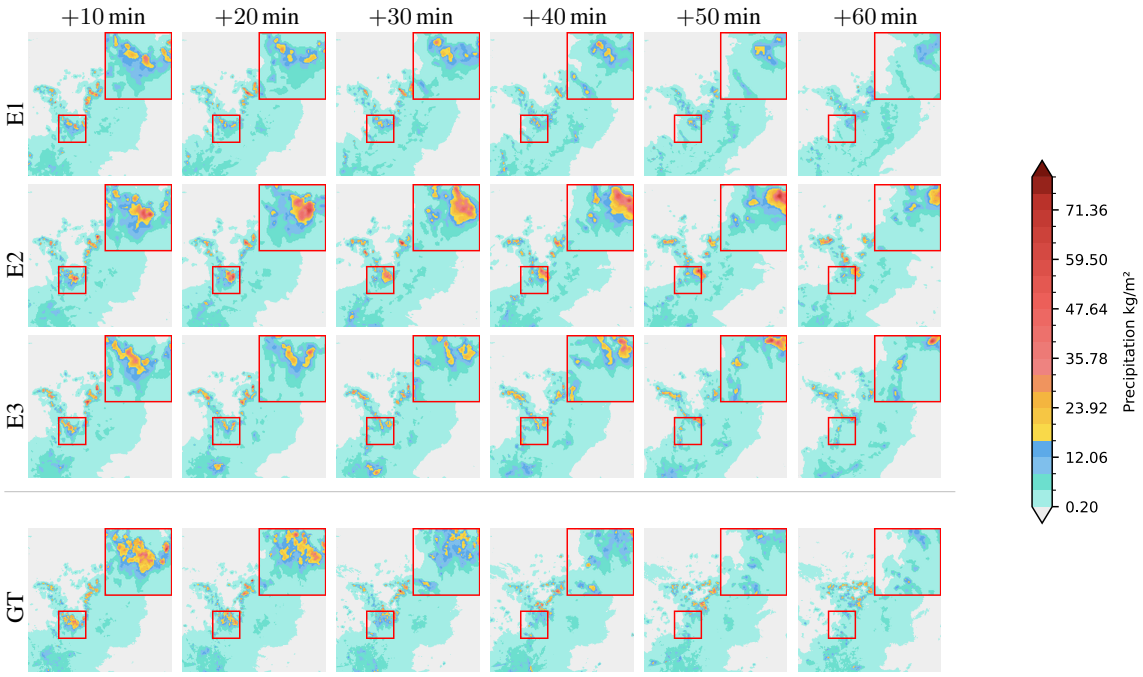


Figure S35. Qualitative result for our L-LSM with *T-reg.* first-stage. The red rectangle highlights a punch-in for better visualization of ensemble differences in details. Best viewed zoomed in.

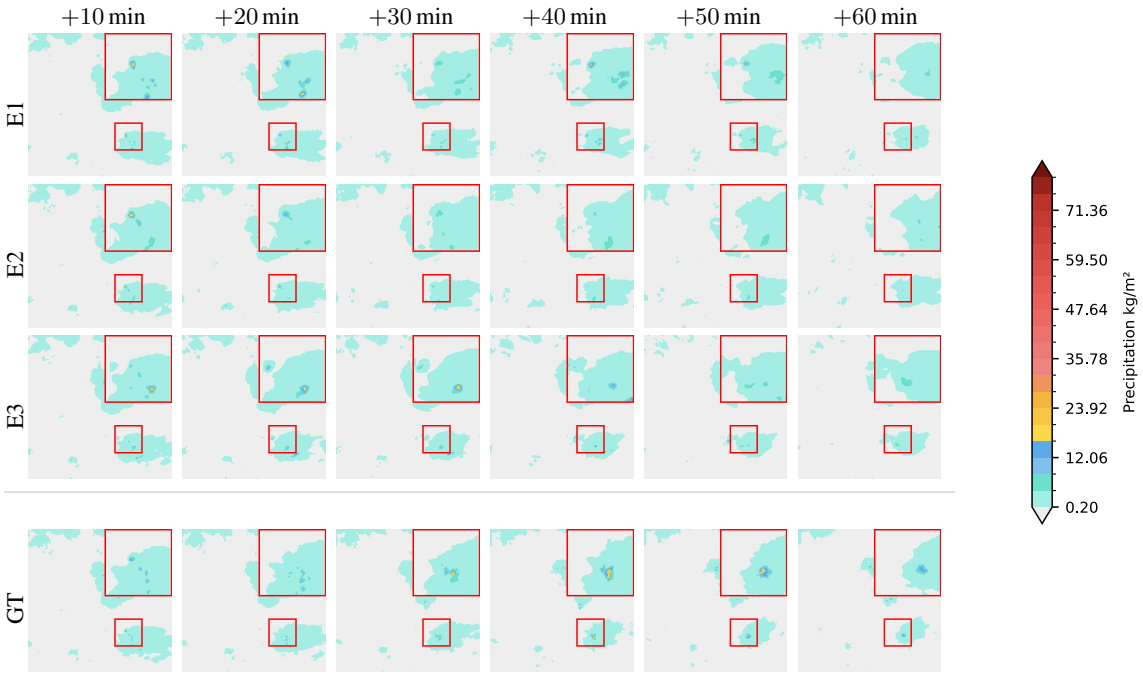


Figure S36. Qualitative result for our L-LSM with *T-reg.* first-stage. The red rectangle highlights a punch-in for better visualization of ensemble differences in details. Best viewed zoomed in.

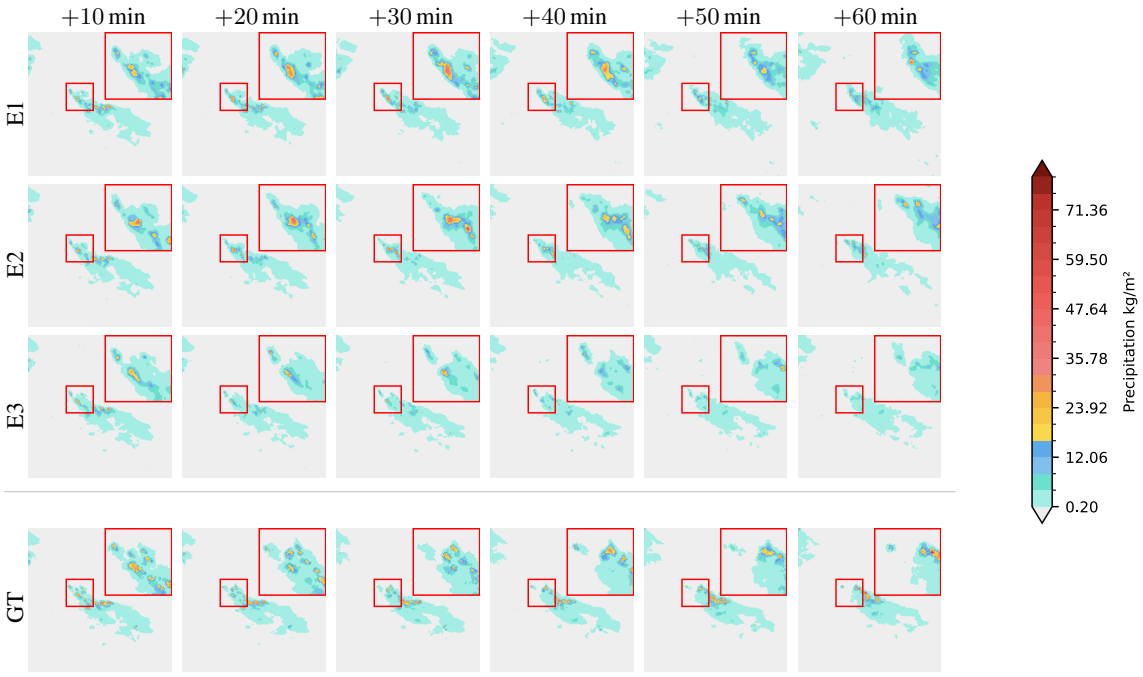


Figure S37. Qualitative result for our L-LSM with *T-reg.* first-stage. The red rectangle highlights a punch-in for better visualization of ensemble differences in details. Best viewed zoomed in.