PRECMAMBA: STATE-SPACE FORECASTING AND FLOW-MATCHING ENSEMBLES FOR EXTREME PRECIPITATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Precipitation nowcasting provides essential short-range forecasts that serve both meteorological science and everyday public safety. Deep learning models for precipitation nowcasting are commonly trained with pixel-wise regression objectives. On sparse, heavy-tailed precipitation fields, conventional objectives tend to produce blurred predictions and suppress the heavy-rain events that matter most. We introduce PRECMAMBA, a two-stage precipitation forecasting framework in which the deterministic stage learns more than the centre of the intensity distribution, letting the generative stage focus on the remaining uncertainty. The first stage employs a Mamba-based deterministic backbone to model the large-scale spatio-temporal evolution of precipitation, trained with a multi-quantile pinball loss that penalises the underprediction of high-intensity events. The second stage models the residual around the deterministic forecast with conditional flow matching, generating sharp and diverse ensemble members in a few integration steps. Experiments on SEVIR, MeteoNet, Shanghai Radar and CIKM show that PRECMAMBA achieves the highest critical success index on all four benchmarks, and on SEVIR it is also best at the most extreme precipitation events.

1 INTRODUCTION

Flash floods and severe convective storms unfold within hours, and warnings against them rely on precipitation nowcasting, the short-range prediction of rainfall from recent radar observations (Ravuri et al., 2021; Zhang et al., 2023). The events that matter most are also the rarest, as they are intense, localised cells that occupy a small fraction of a radar field. Deep learning has substantially improved radar-based nowcasting (Shi et al., 2015; Gao et al., 2022b; Yu et al., 2024), yet its forecasts remain least reliable on extreme precipitation, precisely where warnings need them most. In fact, radar fields are sparse and heavy-tailed, while pixel-wise regression losses, when faced with several plausible futures, reward their conditional mean. Such a mean is a safe compromise for light rain but a poor one for heavy rainfall. As uncertainty grows with lead time, the forecast blurs and intense cells dissolve into weak, widespread rain (Ravuri et al., 2021). Since such extreme events occupy only a small fraction of the field, this degradation can be obscured by otherwise strong aggregate performance.

Existing work tackles this problem along three lines. One line changes the training objective, for instance with Fourier-domain losses (Yan et al., 2024) or amplitude–phase disentanglement (Lin et al., 2025), sharpening the forecast yet returning a single prediction, with no measure of its uncertainty. A second line makes the model fully generative (Voleti et al., 2022; Gao et al., 2023), yielding sharp and diverse samples but forcing the model to spend its capacity regenerating large-scale motion that the observations already determine, often at the cost of accuracy. A third line uses a deterministic network for predicting the large-scale evolution, and a generative model adds the residual (Yu et al., 2024; Gong et al., 2024; Wen et al., 2026). However, these approaches have two weaknesses. First, the deterministic stage is still trained as a mean estimator, which underestimates intense precipitation cells, and the generator must recover the entire tail of the distribution. Second, the generator itself is a diffusion model that needs hundreds of sampling steps (Yu et al., 2024), which makes an ensemble of forecasts expensive.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

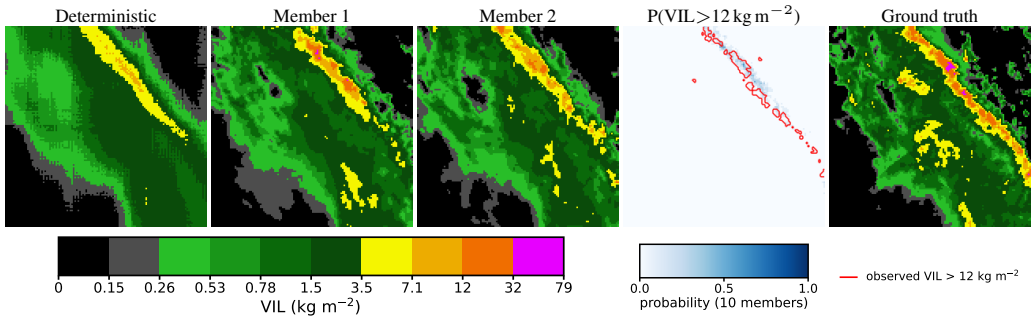

Figure 1: **PRECMAMBA on a storm event at a lead time of 40 minutes.** The deterministic forecast of the first stage captures the large-scale position of the storm, while two ensemble members sampled by the flow-matching stage restore the fine-scale structure and intense cells that the deterministic forecast smooths out. The probability of exceeding 12 kg m^{-2}, estimated from 10 ensemble members, summarises the forecast uncertainty, and the contour marks where the observed field exceeds this threshold. The ground truth uses the same VIL colour scale as the forecasts.

Our key idea is that the deterministic stage should learn more than the centre of the intensity distribution. Exposing the backbone to the heavy-precipitation tail allows the generative stage to focus on the uncertainty that remains around the deterministic forecast. Building on this idea, we introduce PRECMAMBA, a two-stage nowcasting model. Its deterministic stage is a selective state-space backbone (Gu & Dao, 2024; Shams Eddin et al., 2025) that serialises the radar field into a token sequence and models long-range spatio-temporal dependencies at a cost linear in the sequence length. We jointly supervise multiple precipitation quantiles with the pinball loss, whose high-quantile terms penalise the under-prediction of intense precipitation and thus give the shared backbone an explicit training signal on heavy rain. The median prediction then serves as the deterministic forecast used for residual modelling. The generative stage models the remaining uncertainty with conditional flow matching (Lipman et al., 2023), transporting Gaussian noise to the residual along a straight-line probability path and generating each ensemble member in only a few Euler steps. Figure 1 illustrates the two stages on a storm event.

Experiments on SEVIR, MeteoNet, Shanghai Radar and CIKM, four widely used radar nowcasting benchmarks, show that PRECMAMBA achieves the highest Critical Success Index (CSI) on all of them, outperforming state-of-the-art deterministic and generative methods. On SEVIR, it also achieves the highest CSI for the most extreme precipitation events.

In summary, our contributions are (i) a deterministic–generative nowcasting framework in which the deterministic stage learns more than the centre of the intensity distribution, letting the generative stage focus on the remaining uncertainty; (ii) a selective state-space backbone for radar fields and a flow-matching residual generator that samples ensemble members in a few ODE steps; (iii) a new state of the art in CSI on four radar benchmarks, with ablations isolating the backbone, the generator and the training objective.

2 RELATED WORK

Deep Learning for Precipitation Nowcasting. Precipitation nowcasting is usually formulated as a spatio-temporal sequence prediction problem, where sequences of past radar observations are employed to predict future precipitation fields. Early models relied on convolutional recurrent architectures such as ConvLSTM (Shi et al., 2015) and ConvGRU (Ballas et al., 2016), later extended with motion-aware memory in MAU (Chang et al., 2021) and physics-constrained latent dynamics in PhyDNet (Le Guen & Thome, 2020). Other approaches move away from recurrent architectures altogether, from the fully convolutional SimVP (Gao et al., 2022a) to the cuboid attention of Earthformer (Gao et al., 2022b), while STRPM (Chang et al., 2022) employs adversarial and perceptual objectives to improve the visual quality of high-resolution predictions. Despite these architectural advances, pixel-wise regression objectives still attenuate intense, localised structures, and recent approaches explicitly target the preservation of precipitation intensity. NowcastNet (Zhang et al., 2023) combines deterministic physical evolution with conditional generation, FACL (Yan et al., 2024) ad-

dresses the objective itself with Fourier amplitude and correlation losses, and AlphaPre (Lin et al., 2025) disentangles phase and amplitude in the frequency domain.

Probabilistic and Generative Nowcasting. A complementary line of research models a conditional distribution over plausible precipitation sequences rather than predicting a single future. MCVD (Voleti et al., 2022) formulates future-frame prediction as conditional denoising diffusion, PreDiff (Gao et al., 2023) leverages latent diffusion with knowledge-guided sampling, and PercpCast (Feng et al., 2025) refines a posterior-mean forecast with rectified flow to improve structural detail and mitigate intensity decay. Such approaches generate sharper and more diverse predictions than deterministic regression, although directly generating the complete future sequence also requires reproducing components of the evolution that are largely predictable from the observations. Recent work has consequently explored the decomposition of predictable large-scale dynamics from stochastic fine-scale variability. DiffCast (Yu et al., 2024) adds a diffusion-generated residual to the output of an arbitrary deterministic predictor, CasCast (Gong et al., 2024) cascades a deterministic mesoscale forecast with latent probabilistic modelling of small-scale patterns, and DuoCast (Wen et al., 2026) separates low-frequency and high-frequency components with two diffusion models. Our approach is related to this class of deterministic–generative models, but rather than treating the deterministic predictor only as a coarse mean estimate, PRECMAMBA trains it beyond the centre of the intensity distribution and models only the remaining residual with few-step flow matching.

State-Space Models for Spatio-temporal Forecasting. Selective state-space models such as Mamba (Gu & Dao, 2024) allow selective propagation or suppression of information while retaining linear scaling with sequence length, making them particularly amenable to geophysical fields flattened into very long sequences. RiverMamba (Shams Eddin et al., 2025) demonstrates this for hydrological forecasting, and PRECMAMBA adapts its sequence formulation to radar grids. VMU-Diff (Shi et al., 2026) also combines Vision Mamba with residual conditional diffusion for precipitation, though it targets multi-source radar–satellite fusion and uses a coarse-to-fine formulation.

3 METHOD

We introduce PRECMAMBA, a two-stage precipitation nowcasting model that combines a deterministic state-space backbone with a residual generator trained via flow matching, as illustrated in Fig. 2. Precipitation fields are sparse and heavy-tailed, and deterministic models trained with conventional pixel-wise regression objectives tend to average over plausible futures, producing increasingly smooth forecasts in which localised, high-intensity precipitation is progressively attenuated (Ravuri et al., 2021).

Our first stage therefore uses a hierarchical bidirectional Mamba encoder–decoder to capture the predictable large-scale evolution of the precipitation field and produce a deterministic forecast $\boldsymbol{\mu}$, which will be described in Sec. 3.2. The deterministic backbone is trained with a tail-aware objective that encourages it to preserve high-intensity precipitation before residual modelling (Sec. 3.2.1). The second stage models the remaining uncertainty by learning the conditional distribution of the residual $\mathbf{Y} - \boldsymbol{\mu}$, where $\mathbf{Y}$ denotes the future precipitation field, allowing the model to recover fine-scale structure and extreme intensities that are suppressed by the deterministic prediction (Sec. 3.3).

3.1 PROBLEM FORMULATION

We formulate precipitation nowcasting as conditional spatio-temporal sequence prediction (Yu et al., 2024). Let t denote the current time. Given the L_{in} most recent observed radar frames

$$\mathbf{X} = [X_i]_{i=t-L_{\text{in}}}^{t-1} \in \mathbb{R}^{L_{\text{in}} \times H \times W}, \tag{1}$$

our goal is to model the conditional distribution $p(\mathbf{Y} \mid \mathbf{X})$ of the next L_{out} frames

$$\mathbf{Y} = [Y_i]_{i=t}^{t+L_{\text{out}}-1} \in \mathbb{R}^{L_{\text{out}} \times H \times W}. \tag{2}$$

Here, H and W denote the spatial dimensions, $P = HW$ is the number of spatial locations, and each frame is a single-channel radar field. Following Yu et al. (2024), values are normalised to $[0, 1]$ by the sensor ceiling of each product, which places all benchmarks on a common scale, keeps network inputs in a bounded range, and defines the physically valid interval (Sec. 4). We decompose the future sequence as $\mathbf{Y} = \boldsymbol{\mu} + s\mathbf{r}$, where $\boldsymbol{\mu} = \mathcal{B}_{\theta_1}(\mathbf{X})$ is a deterministic forecast produced by

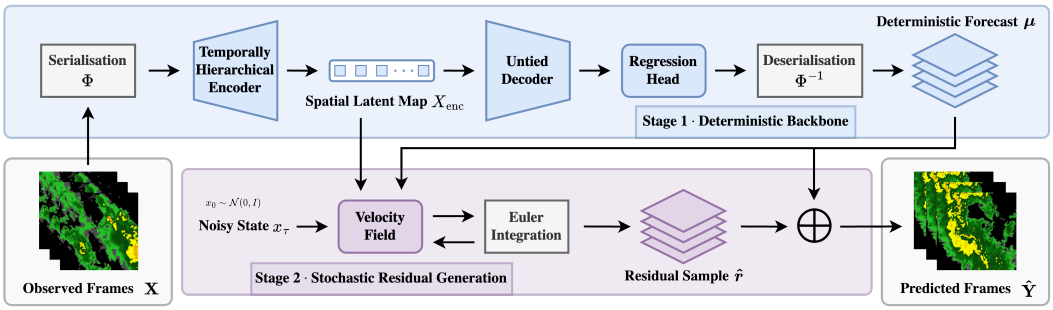

Figure 2: **Overview of PRECMAMBA.** Stage 1 serialises the observed frames with Φ, compresses them into a spatial latent map X_{enc} with a temporally hierarchical encoder, and decodes one output per lead time to produce the deterministic forecast $\boldsymbol{\mu}$. Stage 2 generates an ensemble of predictions as residuals to the deterministic regression. Starting from Gaussian noise x_0, a velocity field conditioned on X_{enc} and $\boldsymbol{\mu}$ is integrated in a few Euler steps, and the resulting residual sample $\hat{\mathbf{r}}$ is rescaled and added to $\boldsymbol{\mu}$.

the backbone with parameters θ_1 (Sec. 3.2), $\mathbf{r}$ denotes the normalised residual variation around this forecast, and s is a dataset-specific scaling factor (Sec. 3.3).

3.2 DETERMINISTIC SPATIO-TEMPORAL BACKBONE

The backbone $\mathcal{B}_{\theta_1}$ maps the observed frames $\mathbf{X}$ to a deterministic forecast $\boldsymbol{\mu} = [\mu_i]_{i=t}^{t+L_{\text{out}}-1}$. It treats the spatial pixels of each precipitation field as a token sequence and is built from three stages, namely serialisation and tokenisation, a temporally hierarchical encoder, and an untied sequential decoder, in which each lead time has its own decoder block and regression head with separate weights, rather than sharing one set of weights across all lead times.

Serialisation and Tokenisation. We adapt the sequence representation of RiverMamba (Shams Eddin et al., 2025) to radar precipitation fields. RiverMamba operates on an irregular global river network and therefore uses multiple alternating scan orders to expose the state-space model to different views of the underlying topology. Radar observations, in contrast, lie on a regular Cartesian grid, where spatial locality can be encoded directly through a fixed serialisation of the 2D field.

We hence define a bijective mapping $\Phi : \{1, \dots, H\} \times \{1, \dots, W\} \to \{1, \dots, P\}$ that converts each radar frame into a one-dimensional token sequence, and denote by π the resulting scan order. Unlike RiverMamba, which alternates scan orders across blocks, we use this single ordering throughout the network. Using a fixed serialisation provides a consistent spatial correspondence throughout the deterministic backbone and allows its latent features to be unambiguously mapped back to the Cartesian grid for conditioning the residual generator. We use a Hilbert space-filling curve, which approximately preserves local neighbourhoods when mapping the 2D grid to one dimension. The inverse mapping Φ^{-1} restores the original spatial geometry after sequence processing. The radar echo of the input is mapped to the hidden dimension K by a shared pointwise embedding with layer normalisation (LN), i.e., $X_{\text{embed}} = \text{LN}(\tanh(\text{Linear}(\mathbf{X}))) \in \mathbb{R}^{L_{\text{in}} \times P \times K}$.

Bidirectional Selective State-Space Blocks. The core processing unit of both encoder and decoder is a bidirectional selective state-space block, applied to the token sequence in the scan order π. Each block applies a bidirectional selective state-space (BiSSM) layer followed by an MLP, both wrapped in pre-norm residual connections. The BiSSM layer averages the outputs of two selective state-space (SSM) layers of Mamba (Gu & Dao, 2024) with separate parameters. The first scans the sequence in the order π, while the second reverses the spatial order within each frame and keeps the temporal order of the frames. Both scans have a cost linear in the sequence length, and their full formulation is given in Sec. B.1.

Temporally Hierarchical Encoder. Rather than processing all input frames at a fixed temporal resolution, the encoder employs a hierarchical merging scheme that progressively reduces the number of temporal states while enlarging the temporal receptive field. At each hierarchy level after

the first, pairs of consecutive temporal features are concatenated along the feature dimension and linearly projected from $2K$ back to K. This halves the temporal resolution while allowing each resulting representation to summarise a broader temporal window. When the number of time steps is odd, the final feature map is zero-padded before merging. A bidirectional selective state-space block is then applied to the resulting spatio-temporal token sequence at the current hierarchy level. At hierarchy level ℓ, the feature tensor has shape $\mathbb{R}^{T_\ell \times P \times K}$, where T_ℓ is the number of temporal states at that level, with $T_1 = L_{\text{in}}$. Before the state-space scan, we reshape it to $\mathbb{R}^{(T_\ell P) \times K}$ by concatenating the serialised spatial tokens of consecutive frames along the sequence dimension. Thus, the BiSSM layer jointly propagates information across space and time within a single spatio-temporal sequence. The final level collapses the temporal dimension to a single spatial latent map $X_{\text{enc}} \in \mathbb{R}^{P \times K}$, which serves as a compact representation of the observed precipitation history for both the deterministic decoder and the residual generator.

Untied Decoder and Regression Heads. The decoder expands the encoded representation X_{enc} into per-lead predictive feature maps. Following RiverMamba (Shams Eddin et al., 2025), we use an *untied* architecture: each lead $j = 1, \ldots, L_{\text{out}}$ has its own bidirectional SSM block $\mathcal{D}_j$ with independent weights, and the blocks are chained, so that each lead is decoded from the representation of the preceding one, $\mathbf{z}_j = \mathcal{D}_j(\mathbf{z}_{j-1})$ with $\mathbf{z}_0 = X_{\text{enc}}$. Assigning separate parameters to each forecast horizon avoids forcing a single transition function to model the dynamics at all lead times and allows the decoder to learn lead-specific transformations. Unlike RiverMamba, which ingests external meteorological forcing at each step, the radar decoder operates solely on the latent representation, since no future covariates are available at inference time. A lightweight MLP head is applied independently to each token of $\mathbf{z}_j$ and outputs one value per quantile level of the training objective (Sec. 3.2.1). A Softplus activation keeps these outputs non-negative, since precipitation cannot be negative. After mapping the tokens back to the grid with Φ^{-1}, the median channel gives the deterministic forecast μ_{t+j-1} at lead j.

3.2.1 Training Objective

A backbone trained with a pixel-wise squared error estimates only the conditional mean of the future field, which attenuates intense precipitation (Sec. 1). Following our key idea, we instead ask the backbone to describe several points of the conditional intensity distribution, including its upper tail. Quantile regression does so without assuming a parametric form of this distribution, which is strongly skewed for precipitation. We therefore train the backbone with the pinball loss over a set of quantile levels $\mathcal{Q}$. For each level $\alpha \in \mathcal{Q}$, the corresponding channel of the regression head predicts an estimate $\hat{q}_\alpha$ of the α-quantile of the target. Writing Ω for the set of all locations for all lead times, $y(p)$ for the target value at $p \in \Omega$, and $e_\alpha(p) = y(p) - \hat{q}_\alpha(p)$, the loss is

$$\mathcal{L}_{\text{pin}} = \frac{1}{|\Omega||\mathcal{Q}|} \sum_{p \in \Omega} \sum_{\alpha \in \mathcal{Q}} \rho_\alpha(e_\alpha(p)), \qquad \rho_\alpha(e) = \begin{cases} \alpha\, e & \text{if } e \geq 0, \\ (\alpha - 1)\, e & \text{if } e < 0. \end{cases} \tag{3}$$

Minimising the expected pinball loss over a prediction q yields the α-quantile of the target distribution. Indeed, the derivative of $\mathbb{E}[\rho_\alpha(y - q)]$ with respect to q is $P(y \leq q) - \alpha$, which vanishes when a fraction α of the targets lies below q. Since the head is conditioned on the observed frames $\mathbf{X}$, each channel thus learns the α-quantile of the target at its location and lead time, just as a squared error would learn the conditional mean. For $\alpha = 0.5$, ρ_α reduces to $0.5\,|e|$, whereas at high levels underprediction is penalised far more heavily than over-prediction, which pushes the shared backbone to encode intense precipitation. Only the median channel is passed to the residual generator as the deterministic forecast $\boldsymbol{\mu}$, while the other channels serve only as training supervision. We compare this objective against the standard mean squared error and a threshold-weighted variant in Sec. 4.

3.3 Stochastic Residual Flow Matching

Despite the tail-aware training objective, a deterministic point forecast cannot represent the full uncertainty of future precipitation evolution. In particular, uncertain fine-scale structures and extreme intensities remain difficult to capture with a single prediction. Hence, following the residual decomposition of DiffCast (Yu et al., 2024), we model the remaining variation as a scaled residual from the deterministic forecast:

$$\mathbf{r} = \text{clip}\big(\mathbf{Y} - \boldsymbol{\mu},\, -1,\, 1\big)/s, \tag{4}$$

where clipping bounds the residual to the range attainable between two valid normalised fields, and s is a dataset-specific scalar equal to the standard deviation of the clipped residual on the validation set. This gives $\mathbf{r}$ approximately unit variance, matching the scale of the Gaussian prior, so that the flow does not need to learn a strong contraction from the prior to a much narrower residual distribution.

Conditional Flow Matching. Following conditional flow matching (Lipman et al., 2023; Tong et al., 2024), we define a straight-line probability path between a Gaussian base distribution $x_0 \sim \mathcal{N}(0, I)$ and the normalised residual target $x_1 = \mathbf{r}$, with marginal noise injection for numerical stability:

$$x_\tau = (1-\tau)\, x_0 + \tau\, x_1 + \sigma\epsilon, \quad \tau \sim \mathcal{U}[0,1],\ \epsilon \sim \mathcal{N}(0, I). \tag{5}$$

Here τ locates a point along the path (one per sample, shared across lead times and locations), ϵ is an independent Gaussian drawn per element, and σ is a small fixed scalar. Together they widen the path into a tube of constant radius. At $\sigma = 0$, it reduces to the Dirac density corresponding to the path of rectified flow (Liu et al., 2023). Being constant in τ, it leaves $(x_1 - x_0)$ as the exact conditional velocity.

A neural velocity field v_{θ_2} is trained to predict the transport direction $(x_1 - x_0)$ at any point along this path, conditioned on the encoder features $\mathbf{c} = X_{\text{enc}}$ and the deterministic forecast $\boldsymbol{\mu}$. The flow-matching objective is

$$\mathcal{L}_{\text{flow}}(\theta_2) = \mathbb{E}_{\tau, x_0, x_1, \epsilon} \left\| v_{\theta_2}(x_\tau, \tau;\, \mathbf{c}, \boldsymbol{\mu}) - (x_1 - x_0) \right\|_2^2. \tag{6}$$

Velocity Network and Dual Conditioning. The velocity field v_{θ_2} receives the noisy state x_τ concatenated with two conditioning signals: the encoder feature map $\mathbf{c}$, which summarises the observed history, and the deterministic forecast $\boldsymbol{\mu}$. The two signals provide complementary information: $\mathbf{c}$ retains context from the observed radar sequence, whereas $\boldsymbol{\mu}$ anchors residual generation to the predicted future precipitation structure. The flow time τ is embedded sinusoidally (Ho et al., 2020) and injected into every block as a learned scale and shift. We implement the velocity network as a convolutional 2D U-Net (Ronneberger et al., 2015) that treats the L_{out} lead times as feature channels and operates directly on the spatial grid. Since the encoder representation $\mathbf{c}$ is produced in the serialised domain, we first map it back to the Cartesian grid using Φ^{-1} before concatenating it with x_τ and $\boldsymbol{\mu}$. A sequential alternative built from the same state-space blocks as the backbone is described and evaluated in Sec. A.5.

Two-Stage Training. We train the deterministic backbone and the residual generator sequentially, for two reasons. First, the residual target $\mathbf{r}$ is defined relative to $\boldsymbol{\mu}$, and its scale s is computed from the trained backbone, so a backbone that keeps changing would turn $\mathbf{r}$ into a moving target. Second, in joint training, the gradient of $\mathcal{L}_{\text{flow}}$ that reaches the encoder is about three orders of magnitude larger than that of the deterministic loss, and would alter the deterministic forecast for reasons unrelated to precipitation dynamics. We therefore optimise $\mathcal{B}_{\theta_1}$ first, and then freeze its parameters while training the flow model, so that $\boldsymbol{\mu}$ and $\mathbf{c}$ act as fixed conditioning signals.

3.4 Inference

Since the second stage models a distribution over residuals rather than a single value, sampling it several times yields an ensemble of forecasts that represents the uncertainty of the future precipitation field. At inference time, the deterministic backbone is evaluated once to obtain $\boldsymbol{\mu}$ and $\mathbf{c}$. For each of M ensemble members, we draw an independent initial sample $x_0^{(m)} \sim \mathcal{N}(0, I)$ and integrate

$$\frac{\mathrm{d}x_\tau}{\mathrm{d}\tau} = v_{\theta_2}(x_\tau, \tau; \mathbf{c}, \boldsymbol{\mu}) \tag{7}$$

from $\tau = 0$ to $\tau = 1$ with n explicit Euler steps. With $\Delta\tau = 1/n$ and $\tau_i = i\Delta\tau$, the update is

$$x_{\tau_{i+1}} = x_{\tau_i} + \Delta\tau\, v_{\theta_2}(x_{\tau_i}, \tau_i; \mathbf{c}, \boldsymbol{\mu}), \quad i = 0, \ldots, n-1. \tag{8}$$

The terminal state x_{τ_n} defines a normalised residual sample $\hat{\mathbf{r}}^{(m)}$. Each residual is mapped back to the original scale and added to the deterministic forecast, i.e.,

$$\hat{\mathbf{Y}}^{(m)} = \text{clip}\left(\boldsymbol{\mu} + s\,\hat{\mathbf{r}}^{(m)},\, 0,\, 1\right), \qquad m = 1, \ldots, M, \tag{9}$$

where clipping enforces the physical bounds of the normalised radar field.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTING

Datasets. We evaluate on the four radar benchmarks used by DiffCast (Yu et al., 2024). SEVIR (Veillette et al., 2020) contains 20,393 storm events of vertically integrated liquid (VIL) over the continental United States, each 4 h long and covering 384×384 km at 1 km and 5-minute resolution. MeteoNet (Larvor et al., 2020) provides three years of radar observations over a 550×550 km area of France. Shanghai Radar (Ma et al., 2022) consists of volume scans at approximately 6-minute intervals from 2015 to 2018, covering 501×501 km around Pudong, Shanghai. CIKM (Shenzhen Meteorological Bureau and Alibaba Group, 2017) contains precipitation events over a 101×101 km area of Guangdong, China, each with 15 radar echo maps at 6-minute intervals.

Data preprocessing. We follow the preprocessing of DiffCast (Yu et al., 2024). The continuous sequences of MeteoNet and Shanghai Radar are separated into discrete events, and all fields are resampled to 128×128 and normalised to $[0, 1]$ by the sensor ceiling of each product. Metrics are computed after mapping the values back to the original data units of each benchmark, in which the evaluation thresholds are defined. We set $L_{\text{in}} = 5$ and $L_{\text{out}} = 20$ throughout, except on CIKM, where $L_{\text{out}} = 10$ is the longest horizon its 15-frame sequences permit.

Evaluation. We report the Critical Success Index (CSI) and the Heidke Skill Score (HSS) at the per-dataset intensity thresholds Λ of DiffCast. For each threshold λ and lead j, true positives TP, false negatives FN and false positives FP are accumulated over the test set, and the scores are averaged over leads and thresholds:

$$\text{CSI}_{\lambda,j} = \frac{\text{TP}_{\lambda,j}}{\text{TP}_{\lambda,j} + \text{FN}_{\lambda,j} + \text{FP}_{\lambda,j}}, \qquad \text{CSI-M} = \frac{1}{|\Lambda|\, L_{\text{out}}} \sum_{\lambda \in \Lambda} \sum_{j=1}^{L_{\text{out}}} \text{CSI}_{\lambda,j}, \tag{10}$$

following the evaluation code of DiffCast (Yu et al., 2024). HSS is computed from the same counts together with the true negatives TN, and aggregated in the same way. Because pixel-wise CSI penalises a correctly forecast but slightly displaced storm twice, once as a miss and once as a false alarm, we additionally report CSI after max-pooling the forecast and target with kernel sizes 4 and 16 (CSI-pool4 and CSI-pool16). We report SSIM (Wang et al., 2004) and LPIPS (Zhang et al., 2018) to measure the visual quality of the forecasts. All metrics for PRECMAMBA are computed on a single ensemble member for fair comparison with previous works.

Implementation details. The backbone uses hidden dimension $K = 192$, a state dimension $N = 8$ in every SSM layer, an MLP expansion ratio of 2 in every block, and four encoder levels for $L_{\text{in}} = 5$. For SEVIR, we use $K = 384$ since it is the largest dataset. It is trained with Adam (Kingma & Ba, 2015) and a cosine learning-rate schedule from 10^{-3} to 10^{-5}, with a batch size of 4 per GPU and gradient clipping at norm 100, and we keep the checkpoint with the best validation CSI. The pinball loss uses the quantile levels $\mathcal{Q} = \{0.1, 0.5, 0.9, 0.95, 0.99\}$. Since the training splits differ in size by more than a factor of thirty, we set the number of epochs per dataset so that each backbone receives a similar number of optimiser steps.

The residual generator is trained afterwards with the backbone frozen, using Adam with a cosine schedule from 2×10^{-4} to 10^{-6}, batch size 4 per GPU, and $\sigma = 0.01$ in Eq. (5). Following common practice for diffusion and flow-matching models (Ho et al., 2020), we keep an exponential moving average (EMA) of the weights with decay 0.999, updated every 10 steps, and use it for evaluation, as it smooths the noisy flow-matching gradients and yields more stable samples. At inference, we integrate Eq. (7) with $n = 10$ explicit Euler steps and, following DiffCast, draw $M = 1$ ensemble member for all quantitative results. All models are trained on 4 NVIDIA GH200 GPUs per run.

4.2 EXPERIMENTAL RESULTS

Table 1 compares PRECMAMBA with deterministic, generative and deterministic–generative baselines on the four benchmarks, where CSI denotes the averaged CSI-M of Eq. (10). Baseline results are taken from the published numbers of DuoCast (Wen et al., 2026) for CSI, HSS and SSIM, and of DiffCast (Yu et al., 2024) for CSI-pool4, CSI-pool16 and LPIPS, as well as for all metrics of ConvGRU, MCVD, STRPM and DiffCast. For DiffCast, we report its best backbone on each dataset,

Table 1: Results on four radar benchmarks. p4 and p16 denote CSI-pool4 and CSI-pool16. Best results in gold, runners-up in blue. Dashes denote metrics not reported by the source.

	SEVIR						MeteoNet					
	CSI↑	p4↑	p16↑	HSS↑	SSIM↑	LPIPS↓	CSI↑	p4↑	p16↑	HSS↑	SSIM↑	LPIPS↓
ConvGRU	0.2416	0.2554	0.3050	0.2834	0.6532	0.3766	0.3400	0.3578	0.4473	0.4667	0.7832	0.2950
SimVP	0.3108	0.2844	0.3452	0.3924	0.6508	0.3914	0.3351	0.3383	0.4143	0.4573	0.7804	0.3523
Earthformer	0.2892	0.2617	0.2910	0.3665	0.6633	0.4140	0.3205	0.3428	0.4333	0.4491	0.7772	0.3718
MAU	0.3076	0.2566	0.2861	0.3863	0.6505	0.3933	0.3233	0.3304	0.4165	0.4452	0.7897	0.3089
PhyDNet	0.3017	0.2685	0.3005	0.3812	0.6532	0.3785	0.3384	0.3824	0.4986	0.4673	0.7823	0.2941
MCVD	0.2148	0.3020	0.4706	0.2743	0.5265	0.2170	0.2336	0.3841	0.6128	0.3393	0.5414	0.1652
PreDiff	0.2744	0.3041	0.4028	0.3592	0.5881	0.2851	0.2944	0.3854	0.5692	0.4462	0.7085	0.1543
STRPM	0.2512	0.3243	0.4959	0.3277	0.6513	0.2577	0.2606	0.4138	0.6882	0.3688	0.5996	0.2004
NowcastNet	0.2791	–	–	0.3512	0.6839	–	0.3427	–	–	0.4751	0.7879	–
CasCast	0.2878	–	–	0.3563	0.5680	–	0.3317	–	–	0.4682	0.7622	–
FACL	0.3161	–	–	0.4033	0.5686	–	0.3612	–	–	0.5006	0.7881	–
DiffCast	0.3077	*0.4122*	*0.5683*	0.4033	0.6354	**0.1812**	0.3512	*0.4930*	*0.7001*	0.4846	0.7887	**0.1244**
AlphaPre	0.3259	–	–	0.4110	*0.6884*	–	0.3824	–	–	0.5164	0.7968	–
DuoCast	*0.3375*	–	–	**0.4318**	0.6827	–	*0.3892*	–	–	**0.5297**	*0.7981*	–
PRECMAMBA	**0.3445**	**0.4256**	**0.6093**	*0.4201*	**0.7209**	*0.2042*	**0.3941**	**0.5482**	**0.7314**	*0.5239*	**0.8179**	*0.1354*

	Shanghai Radar						CIKM					
	CSI↑	p4↑	p16↑	HSS↑	SSIM↑	LPIPS↓	CSI↑	p4↑	p16↑	HSS↑	SSIM↑	LPIPS↓
ConvGRU	0.3612	0.4439	0.5596	0.4899	0.7795	0.2564	0.3092	0.3533	0.4686	0.4007	0.6601	0.3135
SimVP	0.3850	0.4467	0.5603	0.5194	0.7795	0.2984	0.3052	0.3530	0.4677	0.3955	0.6538	0.3134
Earthformer	0.3503	0.4008	0.4863	0.4844	0.7298	0.2564	0.3077	0.3547	0.4927	0.4001	0.6267	0.3857
MAU	0.3983	0.4695	0.5787	0.5346	0.7195	0.2735	0.3039	0.3152	0.4144	0.3928	0.6325	0.3999
PhyDNet	0.3654	0.4552	0.5980	0.4957	0.7751	0.1894	0.3038	0.3442	0.4655	0.3931	0.6541	0.3631
MCVD	0.2872	0.3984	0.5675	0.4036	0.5119	0.2081	0.2513	0.3095	0.4955	0.3294	0.5358	0.2528
PreDiff	0.3504	0.4389	0.5448	0.4991	0.6981	0.1696	0.2842	0.3681	0.5117	0.3481	0.5582	*0.2201*
STRPM	0.3606	0.4944	*0.6783*	0.4931	0.7724	0.1681	0.2984	0.3590	0.5020	0.3870	0.6443	0.2397
NowcastNet	0.3953	–	–	0.5334	0.7902	–	0.2991	–	–	0.3865	*0.6713*	–
CasCast	0.3651	–	–	0.4971	0.7250	–	0.3021	–	–	0.3806	0.5241	–
FACL	0.3940	–	–	0.5308	0.7601	–	0.3105	–	–	0.3996	0.6442	–
DiffCast	0.4089	**0.5212**	0.6658	0.5476	0.7879	*0.1618*	0.3159	*0.3803*	*0.5443*	0.4085	0.6499	0.2205
AlphaPre	0.4178	–	–	0.5534	*0.7951*	–	*0.3194*	–	–	0.4137	0.6568	–
DuoCast	*0.4252*	–	–	*0.5643*	0.7788	–	0.3179	–	–	*0.4217*	0.6659	–
PRECMAMBA	**0.4491**	*0.5102*	**0.6891**	**0.5720**	**0.8109**	**0.1578**	**0.3281**	**0.3923**	**0.5841**	**0.4269**	**0.7031**	**0.2087**

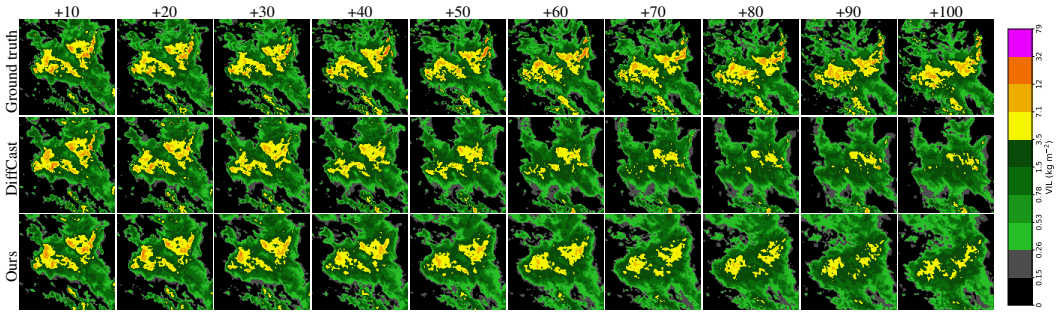

Figure 3: **Qualitative results on SEVIR.** Ground truth and forecasts of DiffCast and PRECMAMBA for one test sequence, shown every 10 minutes up to a lead time of 100 minutes. Both methods are shown as a single sampled forecast. The colour bar gives the vertically integrated liquid (VIL).

namely SimVP on SEVIR, ConvGRU on MeteoNet, and MAU on Shanghai Radar and CIKM, whereas our approach uses the same backbone for all datasets.

PRECMAMBA achieves the best CSI on all four benchmarks. Compared with the strongest competing method, the relative gain is 2.1% on SEVIR, 1.3% on MeteoNet and 5.6% on Shanghai Radar over DuoCast, and 2.7% on CIKM over AlphaPre. The gains are larger at the neighbourhood scale, as PRECMAMBA also achieves the best CSI-pool16 on every benchmark, improving on the second-best method by 7.2% on SEVIR and 7.3% on CIKM. This suggests that the forecasts place precipitation structures correctly at the scale of the pooling window, even where exact pixel alignment is lost. PRECMAMBA is also best in SSIM on all four benchmarks and in HSS on Shanghai Radar and CIKM, while DuoCast retains a small margin in HSS on SEVIR and MeteoNet. At the two highest SEVIR thresholds, which correspond to the most intense precipitation, PRECMAMBA

Table 2: **Ablations on SEVIR.** Left: Training Objective. Right: Backbone and generator. Diffusion denotes the residual diffusion model of DiffCast, and Ours denotes the state-space backbone of Sec. 3.2. SimVP with Diffusion corresponds to DiffCast. Time is the inference time in seconds per forecast. Best results in **bold**.

Loss	CSI↑	p16↑	HSS↑	LPIPS↓
MSE	0.2775	0.2948	0.3283	0.3481
Weighted MSE	0.3115	0.4100	0.3765	0.3813
Pinball	**0.3445**	**0.6093**	**0.4201**	**0.2042**

Backbone	Generator	CSI↑	p16↑	HSS↑	Time↓
SimVP	Diffusion	0.3077	0.5683	0.4033	1.23
SimVP	Flow	0.3145	0.5961	0.4154	0.070
Ours	Diffusion	0.2891	0.4412	0.3355	0.895
Ours	Flow	**0.3445**	**0.6093**	**0.4201**	0.079

reaches a CSI of 0.1150 at $\lambda = 219$, against 0.1074 for DuoCast and 0.0582 for DiffCast (Sec. A.1). At $\lambda = 181$, it reaches 0.1759, close to DuoCast with 0.1818 and well above DiffCast with 0.1300. Compared with DiffCast, which uses the same residual decomposition, the CSI thus increases by 35% at $\lambda = 181$ and nearly doubles at $\lambda = 219$.

The perceptual metric shows the trade-off between accuracy and sharpness. Fully generative models such as MCVD and PreDiff often obtain competitive LPIPS but a much lower CSI, since they must regenerate the large-scale motion as well as the fine-scale detail. Deterministic models show the opposite pattern, with reasonable CSI but typically the worst LPIPS, as their forecasts blur. PRECMAMBA combines the highest CSI with an LPIPS that is best on Shanghai Radar and CIKM and second only to DiffCast on SEVIR and MeteoNet. Since all PRECMAMBA metrics are computed on a single ensemble member, as for DiffCast, this comparison reflects the quality of individual ensemble members rather than of an averaged prediction. Figure 3 shows a SEVIR test sequence.

4.3 ABLATION STUDIES

We ablate the main components of PRECMAMBA on SEVIR, with further ablations in the Appendix.

Training Objective. Table 2 (l.) compares the pinball loss with the standard MSE and the threshold-weighted MSE of Sec. B.2. Moving away from a pure mean estimator improves all hit-based scores, and the pinball loss improves them most, raising CSI by 0.067 over the MSE and by 0.033 over the threshold-weighted MSE. The gap is largest at the neighbourhood scale, where CSI-pool16 increases from 0.295 with the MSE and 0.410 with the threshold-weighted MSE to 0.609. These results support our key idea that the deterministic stage should learn more than the centre of the intensity distribution, and show that supervising the upper quantiles directly is more effective than re-weighting a squared error.

Backbone and Generator. Table 2 (r.) combines two backbones, SimVP and our state-space backbone, with two residual generators, the diffusion model of DiffCast and our flow-matching model. SimVP with diffusion therefore corresponds to DiffCast. Replacing diffusion with flow matching improves both backbones and reduces the inference time by more than an order of magnitude, from 1.23 to 0.070 on SimVP and from 0.895 to 0.079 on our backbone. The generator also matters on its own, as adding it to our backbone raises the CSI from 0.2830 to 0.3445 (Sec. A.3). Diffusion, in contrast, adds little to our backbone and even lowers its HSS (Sec. A.3). This suggests that the two components are complementary, as the flow-matching generator benefits most from a backbone trained beyond the centre of the intensity distribution. The full model achieves the best scores on all accuracy metrics, while adding only 0.027 s to the inference time of the backbone alone.

5 DISCUSSION

We presented PRECMAMBA, a two-stage nowcasting model in which a selective state-space backbone provides the deterministic forecast and a flow-matching generator models the remaining residual. The backbone is trained with the pinball loss, so it learns more than the centre of the intensity distribution. PRECMAMBA achieves the highest CSI on four radar benchmarks, and the ablations show that both the pinball objective and the flow-matching generator contribute to this result. Replacing diffusion with flow matching also reduces the cost of residual generation by more than an order of magnitude, which makes ensemble forecasts practical. We discuss the limitations of this work and directions for future research in Sec. C.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks that required disclosure. We have not used generative AI tools for generating synthetic data, developing theoretical models, formulating mathematical claims, assisting in proofs, proposing hypotheses, designing experiments, implementing methods, translation, dataset cleaning, data analysis, or interpreting results, and no other required disclosure tasks are applicable to this work. Additionally, we used generative AI tools for polishing the manuscript to improve readability. We have reviewed all AI-assisted work. All AI-generated text was reviewed, revised, and verified by the authors; every technical claim, mathematical formulation, and experimental design is entirely the work of the authors. We take responsibility for the final content of this work, including text, claims or artefacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All architectural details, hyperparameters, training procedures, and dataset preprocessing steps are described in the main text and appendix. We follow the publicly available evaluation protocol and data splits of DiffCast (Yu et al., 2024). We will release the source code and trained checkpoints upon acceptance.

REFERENCES

Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *The International Conference on Learning Representations*, 2016.

Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. MAU: A motion-aware unit for video prediction and beyond. In *The Annual Conference on Neural Information Processing Systems*, 2021.

Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. STRPM: A spatiotemporal residual predictive model for high-resolution video prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Elizabeth E. Ebert. Ability of a poor man's ensemble to predict the probability and distribution of precipitation. *Monthly Weather Review*, 129(10):2461–2480, 2001.

Wenzhi Feng, Xutao Li, Zhe Wu, Kenghong Lin, Demin Yu, Yunming Ye, and Yaowei Wang. Perceptually constrained precipitation nowcasting model. In *The International Conference on Machine Learning*, 2025.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. SimVP: Simpler yet better video prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022a.

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 *The Annual Conference on Neural Information Processing Systems*, 2022b.

Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle C. Maddix, Yi Zhu, Mu Li, and Bernie Wang. PreDiff: Precipitation nowcasting with latent diffusion models. In *The Annual Conference on Neural Information Processing Systems*, 2023.

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 *The International Conference on Machine Learning*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *The Annual Conference on Language Modeling*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *The Annual Conference on Neural Information Processing Systems*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations*, 2015.

Gwennaëlle Larvor, Léa Berthomier, Vincent Chabot, Brice Le Pape, Bruno Pradel, and Lior Pérez. MeteoNet: An open reference weather dataset by Météo-France, 2020. URL https://meteofrance.github.io/meteonet/.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Kenghong Lin, Baoquan Zhang, Demin Yu, Wenzhi Feng, Shidong Chen, Feifan Gao, Xutao Li, and Yunming Ye. AlphaPre: Amplitude-phase disentanglement model for precipitation nowcasting. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The International Conference on Learning Representations*, 2023.

Leiming Ma, Lei Chen, Qian Li, and Yechao Xu. Shanghai-2020 dataset, 2022. URL https://doi.org/10.5281/zenodo.7251972.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *The Annual AAAI Conference on Artificial Intelligence*, 2018.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, Rachel Prudden, Amol Mandhane, Aidan Clark, Andrew Brock, Karen Simonyan, Raia Hadsell, Niall Robinson, Ellen Clancy, Alberto Arribas, and Shakir Mohamed. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021. doi: 10.1038/s41586-021-03854-z.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

Mohamad Hakam Shams Eddin, Yikui Zhang, Stefan Kollet, and Jürgen Gall. RiverMamba: A state space model for global river discharge and flood forecasting. In *The Annual Conference on Neural Information Processing Systems*, 2025.

Shenzhen Meteorological Bureau and Alibaba Group. CIKM AnalytiCup 2017: Short-term quantitative precipitation forecasting challenge. https://www.csee.umbc.edu/conference/cikm/CIKM2017/analytiCup.html, 2017.

Chunlei Shi, Hao Li, Yufeng Zhu, Boyu Liu, Yongchao Feng, Zengliang Zang, Hongbin Wang, Yanlan Yang, and Dan Niu. VMU-Diff: A coarse-to-fine multi-source data fusion framework for precipitation nowcasting, 2026. URL https://arxiv.org/abs/2605.14597.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *The Annual Conference on Neural Information Processing Systems*, 2015.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Mark Veillette, Siddharth Samsi, and Chris Mattioli. SEVIR: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In *The Annual Conference on Neural Information Processing Systems*, 2020.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *The Annual Conference on Neural Information Processing Systems*, 2022.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Penghui Wen, Mengwei He, Patrick Filippi, Na Zhao, Feng Zhang, Thomas Francis Bishop, Zhiyong Wang, and Kun Hu. DuoCast: Duo-probabilistic diffusion for precipitation nowcasting. In *The Annual AAAI Conference on Artificial Intelligence*, 2026.

Chiu-Wai Yan, Shi Quan Foo, Van Hoan Trinh, Dit-Yan Yeung, Ka-Hing Wong, and Wai-Kin Wong. Fourier amplitude and correlation loss: Beyond using L2 loss for skillful precipitation nowcasting. In *The Annual Conference on Neural Information Processing Systems*, 2024.

Demin Yu, Xutao Li, Yunming Ye, Baoquan Zhang, Chuyao Luo, Kuai Dai, Rui Wang, and Xunlai Chen. DiffCast: A unified framework via residual diffusion for precipitation nowcasting. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

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.

APPENDIX

This appendix complements the main paper. Section A reports per-threshold results on SEVIR (Sec. A.1), additional experiments on the serialisation order (Sec. A.2), the deterministic backbone without generator (Sec. A.3), the ensemble size and number of Euler steps (Sec. A.4), the sequential velocity network (Sec. A.5) and further qualitative results (Sec. A.6). Section B details the selective state-space layer (Sec. B.1), the alternative training objectives used in the ablation (Sec. B.2) and the architecture of the velocity network (Sec. B.3). Finally, Sec. C discusses the limitations of this work and directions for future research.

A ADDITIONAL EXPERIMENTS

A.1 PER-THRESHOLD RESULTS ON SEVIR

Table 3 reports the CSI of PRECMAMBA, DuoCast and DiffCast at each SEVIR evaluation threshold, where the two highest thresholds correspond to the most intense precipitation. The upper part lists published results, and DuoCast reports only the two highest thresholds. Since its code is not publicly available, we cannot evaluate it at the other thresholds. To compare against DiffCast at every threshold, we additionally run its released checkpoint with the PhyDNet backbone on the same test windows as our model, shown in the lower part.

A.2 SERIALISATION ORDER

We compare the Hilbert space-filling curve used by default with a standard row-major raster scan. Since the state-space layers process the field as a 1D sequence, the scan order decides which locations are close in the recurrence. A raster scan places vertically adjacent locations W steps apart, whereas a Hilbert curve keeps most spatial neighbours close along the sequence. As shown in Tab. 4, the Hilbert curve improves all hit-based scores, raising CSI from 0.2664 to 0.2830 and HSS from 0.2786 to 0.3579. The raster scan obtains better SSIM and LPIPS, which is consistent with smoother forecasts that score well on structural similarity but miss intense cells.

Table 3: **Per-threshold CSI on SEVIR.** Top: published results, where dashes denote thresholds not reported by the source. Bottom: the released DiffCast + PhyDNet checkpoint run by us on the same 576 test windows as our model, with 250 DDIM steps against our 10 Euler steps. CSI is pooled over windows at each lead and averaged over the 20 leads, and Mean is CSI-M. Best results in **bold**.

Method	$CSI_{16}\uparrow$	$CSI_{74}\uparrow$	$CSI_{133}\uparrow$	$CSI_{160}\uparrow$	$CSI_{181}\uparrow$	$CSI_{219}\uparrow$	Mean↑
DiffCast + SimVP	–	–	–	–	0.1300	0.0582	0.3077
DuoCast	–	–	–	–	**0.1818**	0.1074	0.3375
DiffCast + PhyDNet (250 DDIM steps)	0.7044	0.6091	**0.2289**	0.1200	0.0935	0.0417	0.2996
PRECMAMBA, 10 Euler steps	**0.7123**	**0.6574**	0.2193	**0.1914**	0.1759	**0.1150**	**0.3445**

Table 4: **Serialisation order ablation on SEVIR.** Backbone only, trained with the pinball loss, without the generative stage. Best results in **bold**.

Serialisation	CSI↑	p4↑	p16↑	HSS↑	SSIM↑	LPIPS↓
Raster scan	0.2664	0.2523	0.3651	0.2786	**0.6420**	**0.2988**
Hilbert curve (default)	**0.2830**	**0.3288**	**0.4210**	**0.3579**	0.5861	0.3213

A.3 DETERMINISTIC BACKBONE WITHOUT GENERATOR

We evaluate the deterministic stage alone, which forecasts the median channel $\boldsymbol{\mu}$ without residual generation. On SEVIR, it reaches a CSI of 0.2830, a CSI-pool16 of 0.4210 and an HSS of 0.3579, with an inference time of 0.052 s per forecast. Adding the flow-matching generator raises these scores to 0.3445, 0.6093 and 0.4201 (Tab. 2), at an extra cost of only 0.027 s. Adding the diffusion model of DiffCast instead gives a CSI of 0.2891 and lowers the HSS to 0.3355. The backbone alone also scores below the deterministic baselines of Tab. 1, such as SimVP with a CSI of 0.3108. This is expected, since the median of a right-skewed distribution lies below its mean, so a median forecast underpredicts intense pixels and misses many threshold exceedances. The upper-quantile terms of the pinball loss do not act through $\boldsymbol{\mu}$, but shape the encoder features $\mathbf{c}$ during training, which the generator receives as conditioning together with $\boldsymbol{\mu}$ (Sec. 3.3). We therefore attribute the gain of the full model to the combination of both stages, rather than to the deterministic forecast alone.

A.4 ENSEMBLE SIZE AND NUMBER OF EULER STEPS

We study the two sampling parameters of Sec. 3.4 on CIKM, namely the ensemble size M and the number of Euler steps n (Fig. 4).

Ensemble size. To score an ensemble with CSI, we combine its members into a single field with the probability-matched mean (PMM) (Ebert, 2001). The PMM keeps the spatial pattern of the ensemble mean but restores the intensity distribution of the members, so it does not blur intense cells as a plain average would. A single member reaches a CSI-M of 0.325, above the 0.322 of the deterministic forecast $\boldsymbol{\mu}$. The CSI-M rises to 0.342 with $M = 5$ and to 0.346 with $M = 20$. Five members thus give about 80% of the total gain, while the sampling time grows linearly with M, from 0.05 s to 0.8 s per forecast.

Number of Euler steps. With a single step, the sample is further from the ground truth than the deterministic forecast, with an LPIPS of 0.356 against 0.303. Two steps match the deterministic forecast, and the LPIPS then decreases steadily to 0.209 at $n = 10$. Doubling to $n = 20$ improves it only to 0.202, while the sampling time increases from 0.049 s to 0.081 s. We therefore use $n = 10$ in all experiments, as it gives almost all of the gain at a moderate cost.

A.5 SEQUENTIAL VELOCITY NETWORK

To study the effect of the spatial inductive bias in the generative stage, we additionally implement a sequential velocity network built from the same bidirectional selective state-space blocks as the deterministic backbone. In this variant, the lead times are folded into the channel dimension and the network operates directly on the serialised token sequence. Therefore, $\mathbf{c}$ is processed in its native

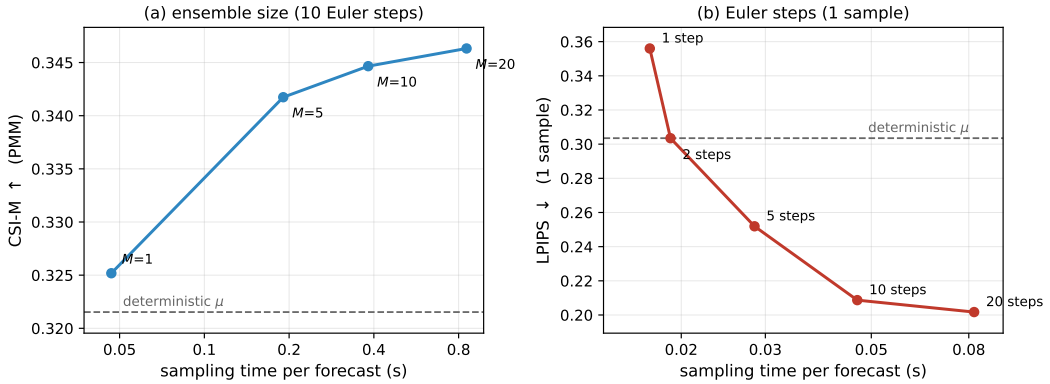

Figure 4: **Sampling parameters on CIKM.** (a) CSI-M of the probability-matched mean of M ensemble members, sampled with $n = 10$ Euler steps. (b) LPIPS of a single member for different numbers of Euler steps n. The dashed line gives the deterministic forecast $\boldsymbol{\mu}$ of the backbone. Times are per forecast, and the horizontal axes are logarithmic.

1D representation, and the predicted velocity is mapped back to the spatial grid using Φ^{-1}. The two variants share the same conditioning signals and flow-matching objective.

Table 5: **Velocity network ablation on SEVIR.** Both variants share the same frozen backbone and residual scale s, and are evaluated on a single sampled forecast. Best results in **bold**.

Velocity network	CSI↑	p4↑	p16↑	HSS↑	SSIM↑	LPIPS↓
2D U-Net (default)	**0.3445**	**0.4256**	**0.6093**	**0.4201**	0.7209	**0.2042**
Sequential SSM	0.3113	0.4009	0.5987	0.4182	**0.7301**	0.3012

Table 5 compares the two variants on SEVIR. The convolutional U-Net improves CSI from 0.3113 to 0.3445 and reduces LPIPS from 0.3012 to 0.2042, while CSI-pool16 and HSS remain close for both variants. The sequential network thus places precipitation at roughly the right scale, but produces less sharp residuals, which is also reflected in its slightly higher SSIM. We attribute this to the nature of the residual, which mostly contains the fine-scale structure that the deterministic forecast cannot represent. A U-Net suits this structure, as its local convolutions operate on the native grid and its skip connections carry fine detail past the bottleneck. The sequential network, in contrast, processes a 1D serialisation, in which some spatially adjacent locations are necessarily far apart along the scan, even with a locality-preserving curve.

A.6 Additional Qualitative Results

Figures 5 to 8 show further test sequences from the four benchmarks, comparing the ground truth with single sampled forecasts of DiffCast and PrecMamba. As in Fig. 3, PrecMamba generally preserves intense cells at longer lead times, while DiffCast progressively weakens them. Figure 9 shows five ensemble members for three SEVIR sequences at lead times of 40, 60 and 100 minutes, together with the deterministic forecast and the exceedance probability estimated from 10 members.

B Additional Formulation Details

B.1 Selective State-Space Layer Details

Bidirectional Selective State-Space Blocks. The core processing unit of both encoder and decoder is a bidirectional selective state-space block, applied to the token sequence in the scan order π. Each block applies a bidirectional selective state-space (BiSSM) layer followed by an MLP, both wrapped in pre-norm residual connections:

$$u = x + \mathrm{BiSSM}\big(\mathrm{LN}(x)\big), \qquad x' = u + \mathrm{MLP}\big(\mathrm{LN}(u)\big). \tag{11}$$

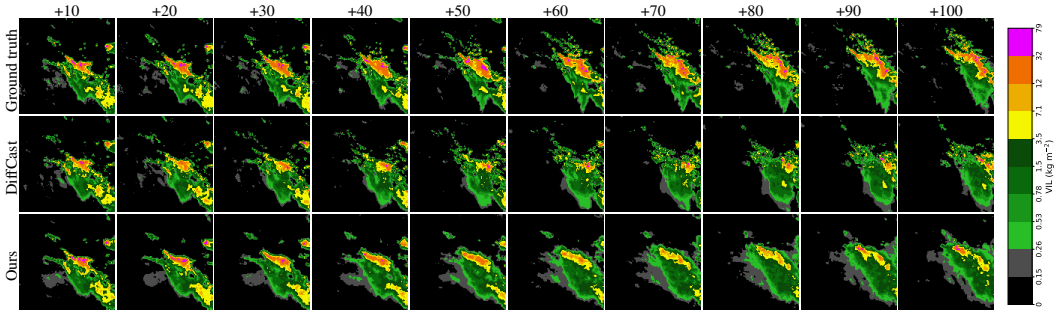

Figure 5: **Qualitative results on SEVIR.** Ground truth and forecasts of DiffCast and PRECMAMBA for one test sequence, shown every 10 minutes up to a lead time of 100 minutes. Both methods are shown as a single sampled forecast. The colour bar gives the vertically integrated liquid (VIL).

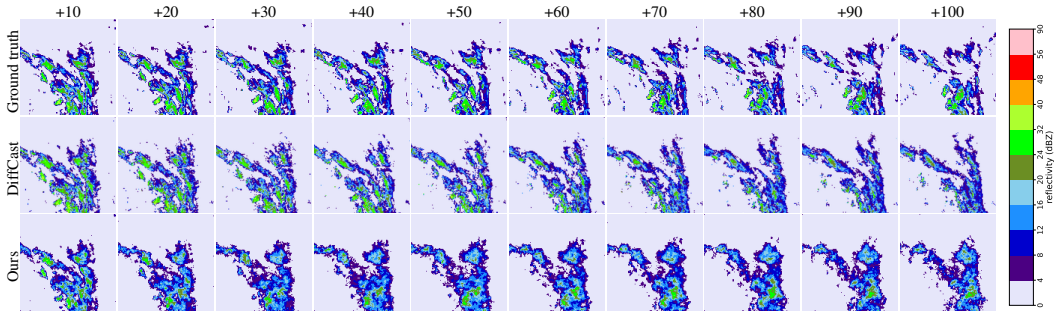

Figure 6: **Qualitative results on MeteoNet.** Ground truth and forecasts of DiffCast and PRECMAMBA for one test sequence, shown every 10 minutes up to a lead time of 100 minutes. Both methods are shown as a single sampled forecast. The colour bar gives the radar reflectivity.

The BiSSM layer is built from the gated selective state-space (SSM) layer of Mamba (Gu & Dao, 2024), without the skip term D, denoted $\mathrm{SSM}(\cdot)$, which sweeps a recurrent latent state along π. Let n denote a token's position in the scan, d a feature channel, and $\tilde{u}_n$ the projected input token (defined below). Each channel d carries a hidden state $h_n^{(d)} \in \mathbb{R}^N$, where N is the state dimension, which is updated and read out as

$$h_n^{(d)} = \overline{A}_n^{(d)} \odot h_{n-1}^{(d)} + \overline{B}_n^{(d)}\, \tilde{u}_n^{(d)}, \qquad o_n^{(d)} = C_n^\top h_n^{(d)}. \tag{12}$$

Here, $\overline{A}_n^{(d)} \in \mathbb{R}^N$ is the discretised state transition, $\overline{B}_n^{(d)} \in \mathbb{R}^N$ the discretised input projection, and $C_n \in \mathbb{R}^N$ the output projection. The continuous state matrix $A^{(d)}$ is diagonal and learned per channel, so the update is an element-wise product instead of a matrix multiplication.

What makes the sweep selective is that the transition is not fixed but set by the token itself. A positive step size $\Delta_n^{(d)}$ and the projections $B_n, C_n \in \mathbb{R}^N$ are predicted from $\tilde{u}_n$ by a learned function f, and the discretised parameters follow from them:

$$\overline{A}_n^{(d)} = \exp\!\big(\Delta_n^{(d)} A^{(d)}\big), \qquad \overline{B}_n^{(d)} = \Delta_n^{(d)} B_n, \qquad \Delta_n,\, B_n,\, C_n = f(\tilde{u}_n). \tag{13}$$

The form of f, the discretisation it approximates and the gating are given below.

A single scan propagates information only forward along π. This directional bias is undesirable for spatial precipitation fields, where dependencies are not intrinsically oriented along the arbitrary serialisation order. The BiSSM layer therefore combines two SSM layers with separate parameters, $\mathrm{SSM}_{\rightarrow}$ and $\mathrm{SSM}_{\leftarrow}$. Both process the same temporally ordered sequence of frames, but the second reverses the spatial ordering π within each frame while leaving the temporal frame order unchanged:

$$\mathrm{BiSSM}(x) = \tfrac{1}{2}\big(\mathrm{SSM}_{\rightarrow}(x) + \mathrm{flip}_P(\mathrm{SSM}_{\leftarrow}(\mathrm{flip}_P(x)))\big). \tag{14}$$

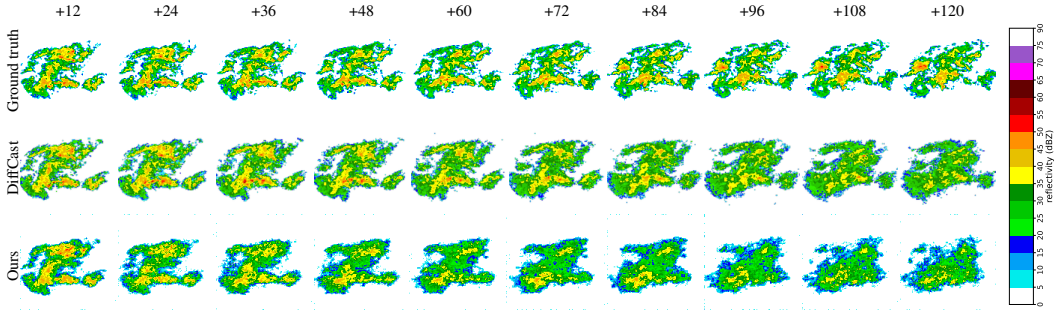

Figure 7: **Qualitative results on Shanghai Radar.** Ground truth and forecasts of DiffCast and PRECMAMBA for one test sequence, shown every 12 minutes up to a lead time of 120 minutes. Both methods are shown as a single sampled forecast. The colour bar gives the radar echo intensity.

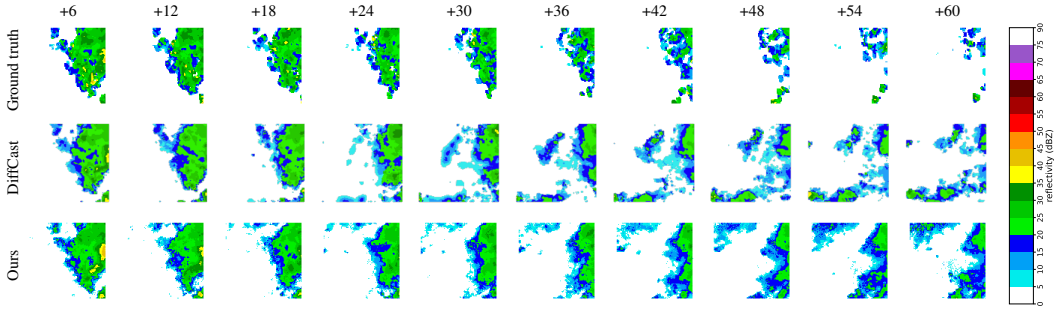

Figure 8: **Qualitative results on CIKM.** Ground truth and forecasts of DiffCast and PRECMAMBA for one test sequence, shown every 6 minutes up to a lead time of 60 minutes. Both methods are shown as a single sampled forecast. The colour bar gives the radar echo intensity.

Input projection. Let $x_n \in \mathbb{R}^K$ be the layer-normalised input token at scan position n. A linear map projects it to twice the inner width,

$$[\, v_n;\, g_n \,] = W_{\text{in}}\, x_n, \qquad v_n, g_n \in \mathbb{R}^K. \tag{15}$$

The sequence v passes through a depthwise causal 1D convolution of kernel size 3 along the token axis, followed by a SiLU activation, giving the SSM input $\tilde{u}_n = \text{SiLU}\big(\text{Conv1D}(v)_n\big)$. The second half, g_n, is kept for gating.

Selection function. The function f in Eq. (13) is a shared low-rank projection W_x applied to $\tilde{u}_n$. Its output provides B_n and C_n directly, and the step size through a second projection:

$$\Delta_n = \text{softplus}\big(W_\Delta\, W_x\, \tilde{u}_n + b_\Delta\big), \tag{16}$$

where the softplus keeps Δ_n strictly positive.

Discretisation. The exact zero-order-hold discretisation of the continuous system gives

$$\overline{A}_n^{(d)} = \exp\big(\Delta_n^{(d)} A^{(d)}\big), \qquad \overline{B}_n^{(d)} = \big(\Delta_n^{(d)} A^{(d)}\big)^{-1}\big(\exp(\Delta_n^{(d)} A^{(d)}) - I\big)\, \Delta_n^{(d)} B_n. \tag{17}$$

Following the Mamba implementation, we use the first-order approximation $\overline{B}_n^{(d)} = \Delta_n^{(d)} B_n$ for the input term, as in Eq. (13).

Gating and output projection. The scan output o_n (Eq. (12)) is gated by g_n and projected back to the hidden width:

$$\text{SSM}(x)_n = W_{\text{out}}\big(o_n \odot \text{SiLU}(g_n)\big). \tag{18}$$

B.2 ALTERNATIVE TRAINING OBJECTIVES

We compare the pinball loss used by default (Sec. 3.2.1) with two alternative objectives for the deterministic backbone. Writing Ω for the set of all locations for all lead times, and y and μ for the target and forecast values at a position, the first is the standard mean squared error.

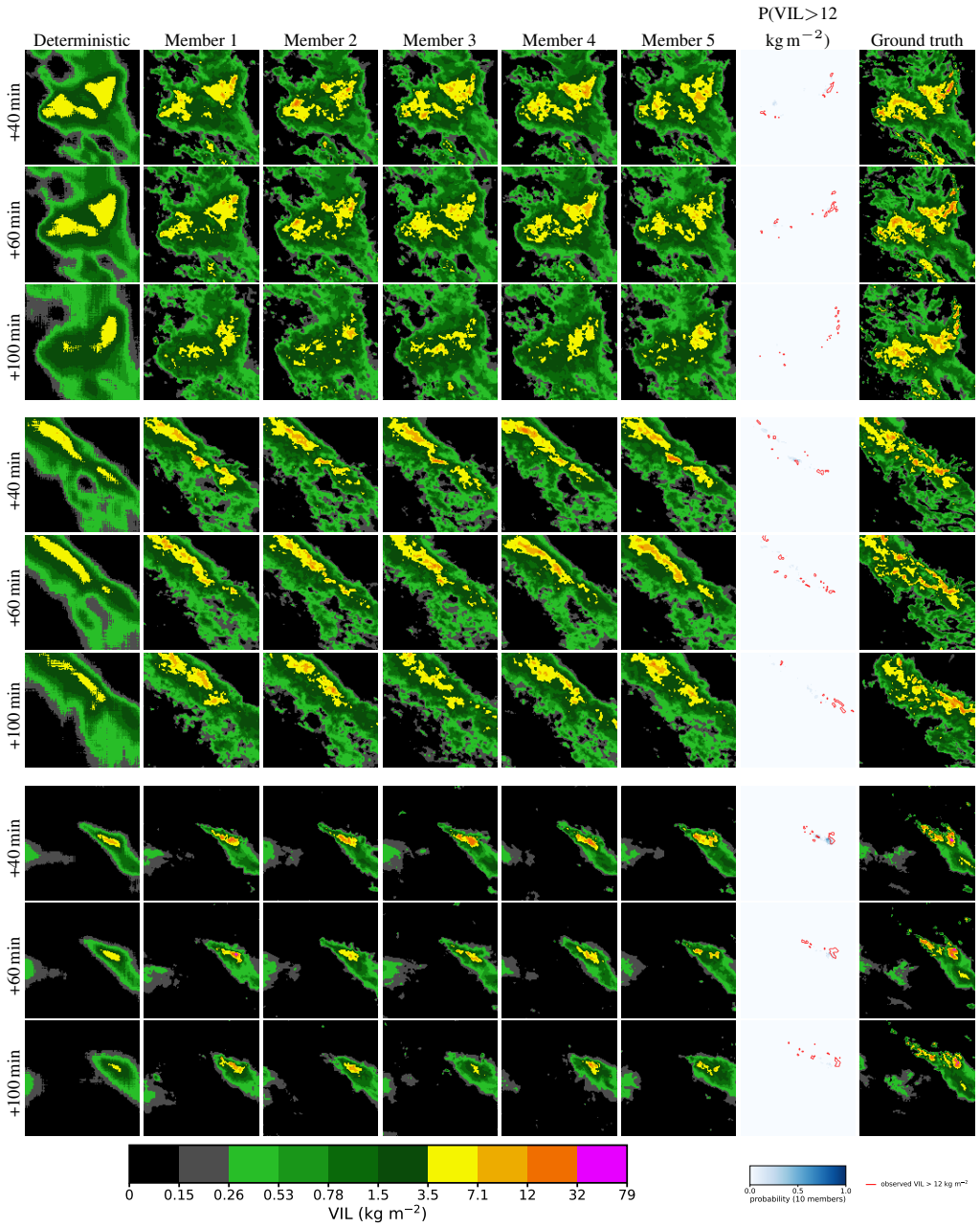

Figure 9: **Ensemble members of PRECMAMBA on SEVIR.** Deterministic forecast, five ensemble members sampled from independent initial noise, and the probability of exceeding 12 kg m^{-2} estimated from 10 members, at lead times of 40, 60 and 100 minutes. The contour marks where the observed field exceeds the threshold, and the ground truth uses the same VIL colour scale as the forecasts.

The second is a *threshold-weighted* squared error that raises the cost of missing intense precipitation events. With $\nu(p) = \sum_k \mathbb{1}[\, y(p) > \lambda_k \,]$ counting how many intensity thresholds $\{\lambda_k\}$ a target value exceeds,

$$\mathcal{L}_{\text{wMSE}} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \gamma^{\,\nu(p)} \, (\mu(p) - y(p))^2, \qquad \gamma > 1. \tag{19}$$

The denominator is the unweighted count $|\Omega|$ rather than the sum of weights, so pixels with intense precipitation contribute more to the total loss. The thresholds $\{\lambda_k\}$ are the standard evaluation thresholds published with each benchmark. We employ $\gamma = 3$.

B.3 Velocity Network Architecture

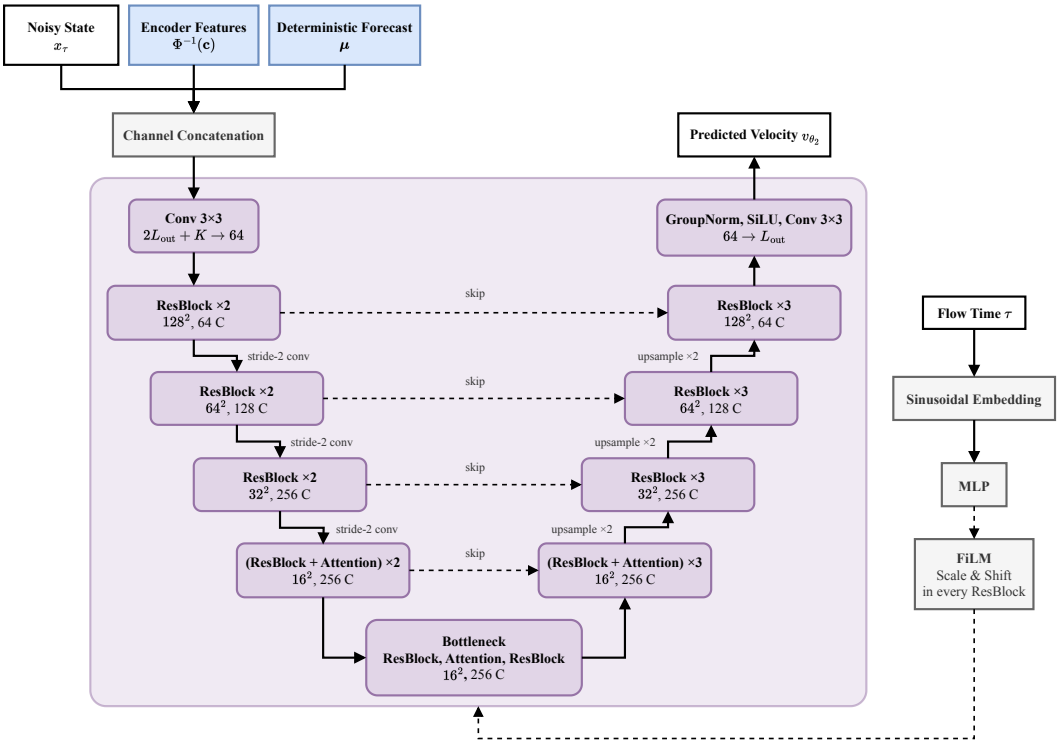

Figure 10: **Velocity network of PrecMamba.** A convolutional U-Net predicts the velocity v_{θ_2} from the noisy state x_τ, the encoder features $\Phi^{-1}(\mathbf{c})$ and the deterministic forecast $\boldsymbol{\mu}$, concatenated along the channel dimension. The encoder halves the resolution three times, from 128^2 to 16^2, and self-attention is used only at 16^2. Horizontal dashed arrows denote skip connections, and C denotes the number of channels. The flow time τ is embedded sinusoidally and modulates every residual block through a learned scale and shift.

Figure 10 shows the convolutional U-Net (Ronneberger et al., 2015) that implements the velocity field v_{θ_2} of Sec. 3.3. Its input concatenates three tensors along the channel dimension, namely the noisy state x_τ and the deterministic forecast $\boldsymbol{\mu}$ with L_{out} channels each, and the encoder features $\Phi^{-1}(\mathbf{c})$ with K channels. A 3×3 convolution maps these $2L_{\text{out}} + K$ channels to 64.

Encoder and bottleneck. The encoder has four resolution levels, from 128^2 to 16^2, with 64, 128, 256 and 256 channels. Each level applies two residual blocks, and a stride-2 convolution halves the resolution between levels. At 16^2, each residual block is followed by a self-attention layer, since the number of positions is smallest at this level. The bottleneck applies a residual block, a self-attention layer and a second residual block at 16^2 with 256 channels.

Decoder and output. The decoder mirrors the encoder with three residual blocks per level, and upsamples the features by a factor of two between levels. Skip connections pass the features of each encoder level to the decoder level at the same resolution. A final group normalisation, a SiLU activation and a 3×3 convolution map the 64 channels to L_{out}, which gives one velocity channel per lead time.

Time conditioning. The flow time τ is embedded sinusoidally (Ho et al., 2020) and passed through an MLP. From this embedding, each residual block predicts a per-channel scale and shift that modu-

late its features (Perez et al., 2018). The input convolution and the output layers are not conditioned on τ.

C Limitations and Future Work

Limitations. Following DiffCast (Yu et al., 2024), we evaluate a single sampled forecast, which allows a direct comparison with published results. On SEVIR, the median forecast of the backbone alone scores below deterministic baselines (Sec. A.3), so the benefit of the quantile supervision appears only together with the generator. Finally, we use common preprocessing at 128×128 pixels, which is coarser than SEVIR's native resolution and partly smooths small convective cells before training.

Future Work. The generator is currently conditioned only on the median forecast $\boldsymbol{\mu}$ and the encoder features $\mathbf{c}$, while the upper-quantile outputs of the head are discarded at inference. Passing these outputs to the generator as additional conditioning could give it a direct signal on where intense precipitation is likely. We also plan to evaluate PrecMamba at native radar resolution and longer temporal horizons.