# FLOWCAST: ADVANCING PRECIPITATION NOWCASTING WITH CONDITIONAL FLOW MATCHING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Radar-based precipitation nowcasting, the task of forecasting short-term precipitation fields from previous radar images, is a critical problem for flood risk management and decision-making. While deep learning has substantially advanced this field, two challenges remain fundamental: the uncertainty of atmospheric dynamics and the efficient modeling of high-dimensional data. Diffusion models have shown strong promise by producing sharp, reliable forecasts, but their iterative sampling process is computationally prohibitive for time-critical applications. We introduce FlowCast, the first model to apply Conditional Flow Matching (CFM) to precipitation nowcasting. Unlike diffusion, CFM learns a direct noise-to-data mapping, enabling rapid, high-fidelity sample generation with drastically fewer function evaluations. Our experiments demonstrate that FlowCast establishes a new state-of-the-art in predictive accuracy. A direct comparison further reveals the CFM objective is both more accurate and significantly more efficient than a diffusion objective on the same architecture, maintaining high performance with significantly fewer sampling steps. This work positions CFM as a powerful and practical alternative for high-dimensional spatiotemporal forecasting.

## 1 INTRODUCTION

Accurate and timely short-term precipitation forecasts, or nowcasting, are of paramount importance due to their significant socio-economic impacts, such as issuing flood warnings and managing water resources. Precipitation nowcasting, as defined in this work, involves predicting a sequence of future radar images from historical observations for the immediate future up to a few hours (An et al., 2025). Traditional methods, like Eulerian and Lagrangian persistence (Germann & Zawadzki, 2002), rely on advecting the current precipitation field. However, their simplified physical assumptions limit their ability to capture the complex, non-linear dynamics of atmospheric processes, especially for rapidly evolving weather systems (Prudden et al., 2020).

Deep learning has introduced a paradigm shift in precipitation nowcasting. Deterministic models based on recurrent and transformer architectures learn complex spatiotemporal patterns directly from large volumes of radar data (Prudden et al., 2020; An et al., 2025). While these models outperform traditional methods, optimizing for metrics like Mean Squared Error (MSE) compels them to produce a single, best-guess forecast. This often results in overly smooth predictions at longer lead times, failing to capture the inherent uncertainty in precipitation evolution and underrepresenting high-impact weather events.

To address this, probabilistic generative models have become central to modern nowcasting, aiming to predict a distribution over many plausible futures. Diffusion models (Ho et al., 2020), in particular, have emerged as the state-of-the-art, producing sharp and reliable ensemble forecasts (Gao et al., 2023; Leinonen et al., 2023; Gong et al., 2024). However, this performance comes at a steep price: their reliance on an iterative denoising process, often requiring hundreds of function evaluations for a single forecast, makes them computationally expensive. This high Number of Function Evaluations (NFE) poses a significant barrier to practical application in time-critical scenarios where rapid ensemble generation is crucial.

This work investigates Conditional Flow Matching (CFM) (Lipman et al., 2023; Tong et al., 2024) as a powerful and more efficient alternative designed for rapid sampling. We introduce FlowCast,

the first model to apply CFM to probabilistic precipitation nowcasting. We demonstrate that FlowCast resolves the tension between accuracy and efficiency, establishing a new state-of-the-art by exceeding the performance of leading diffusion models while offering a superior performance-cost trade-off.

Our contributions are summarized as follows:

- We introduce FlowCast, the first application of Conditional Flow Matching to precipitation nowcasting.
- We establish a new state-of-the-art in probabilistic performance on two diverse radar datasets, the benchmark SEVIR dataset (Veillette et al., 2020) and the local ARSO dataset.
- We provide a direct ablation study showing that the CFM objective is both more accurate and more computationally efficient than a diffusion objective on the same architecture, maintaining high performance with substantially fewer sampling steps.

## 2 Related Work

**Deterministic Nowcasting.** Deep learning for precipitation nowcasting has evolved from RNN-based architectures to Transformer-based models. Early work includes ConvLSTM (Shi et al., 2015), extending LSTMs with convolutions for spatiotemporal data, and the PredRNN family (Wang et al., 2017; 2023), which introduced a spatiotemporal memory flow for improved long-range dependency modeling. More recently, Transformer architectures like Earthformer (Gao et al., 2022) and Earthfarseer (Wu et al., 2024) have set new benchmarks by using attention to model complex global dynamics. A common limitation of deterministic models is that they produce overly smooth forecasts when trained with pixel-wise losses (e.g., MSE), as they average over possible futures.

**Probabilistic Nowcasting.** To address uncertainty quantification, probabilistic models have become central to nowcasting. GANs (Ravuri et al., 2021) were an early approach for producing sharp forecasts, but diffusion models (Ho et al., 2020) are now state-of-the-art, offering stable training and high-quality samples. PreDiff (Gao et al., 2023) and LDCast (Leinonen et al., 2023) are prominent latent diffusion models for ensemble forecasting. A notable hybrid is CasCast (Gong et al., 2024), which uses a deterministic model for large-scale patterns and a conditional diffusion model to refine stochastic details, and is currently the state-of-the-art for probabilistic precipitation nowcasting.

**Flow-Based Generative Models.** While diffusion models deliver state-of-the-art results, their iterative sampling is computationally demanding. Continuous Normalizing Flows (CNFs) enable more efficient sampling. Recently, Conditional Flow Matching (CFM) (Lipman et al., 2023) has emerged as a powerful technique for training CNFs, allowing for simulation-free training and rapid generation. It has shown strong performance in computer vision (Tong et al., 2024; Dao et al., 2023; Jin et al., 2025), but its potential for scientific forecasting remains unexplored. To our knowledge, this work is the first to apply CFM to probabilistic precipitation nowcasting.

## 3 Method

Our approach to probabilistic nowcasting is based on Conditional Flow Matching (CFM) within a compressed latent space. This section details our methodology, covering the problem formulation, our latent CFM framework, the model architecture, and the training and sampling procedures.

### 3.1 Task Formulation

Precipitation nowcasting is framed as a video prediction task. Given a sequence of $T_{\text{in}}$ past radar observations, $\mathbf{X}_{\text{past}} = \{x_1, x_2, \ldots, x_{T_{\text{in}}}\}$, where each $x_t \in \mathbb{R}^{H \times W \times C}$ is a radar map, the objective is to generate a probabilistic forecast for the next $T_{\text{out}}$ frames, $\mathbf{X}_{\text{future}} = \{x_{T_{\text{in}}+1}, \ldots, x_{T_{\text{in}}+T_{\text{out}}}\}$.

### 3.2 Latent Conditional Flow Matching

To reduce the high computational cost of generative modeling, we adopt a two-stage approach inspired by latent diffusion models (Rombach et al., 2022). A Variational Autoencoder (VAE) (Kingma et al., 2013) compresses high-dimensional radar frames into low-dimensional latents, which are used to train a generative model in the latent space.

Our generative model is built on the Conditional Flow Matching (CFM) framework (Lipman et al., 2023), which trains a continuous normalizing flow by learning a vector field $v_\theta$ that maps samples from a prior distribution (e.g., Gaussian) to the target data distribution. We use Independent CFM (I-CFM) (Tong et al., 2024), which defines a probability path $p_t(x_t|x_0, x_1)$ as a Gaussian distribution with mean $(1-t)x_0+tx_1$ and a small constant standard deviation $\sigma$. This path interpolates between a noise sample $x_0 \sim \mathcal{N}(0, I)$ and a data sample $x_1$. The corresponding target vector field is their difference, $u_t = x_1 - x_0$. This formulation enables direct, simulation-free training of the model $v_\theta$ by regressing it against this target field.

#### 3.2.1 Frame-wise Autoencoder

To learn a compact latent space, we train a VAE on individual radar frames. The architecture, inspired by Esser et al. (2021), uses a hierarchical encoder $\mathcal{E}$ and decoder $\mathcal{D}$ with residual and self-attention blocks for high-fidelity reconstructions. The VAE is trained with a combination of a L1 reconstruction loss, a KL-divergence regularizer, and a PatchGAN adversarial loss (Isola et al., 2017) to enhance perceptual quality. After training, the VAE's weights are frozen and it is used to encode inputs and decode latent predictions.

#### 3.2.2 FlowCast Architecture

We propose **FlowCast**, which consists of the adaptation of Earthformer-UNet (Gao et al., 2023) for the CFM objective. FlowCast employs a U-Net-like encoder-decoder structure where the core building blocks are Cuboid Attention layers from Earthformer (Gao et al., 2022). This mechanism efficiently processes spatiotemporal data by applying self-attention locally within 3D "cuboids" of the data, capturing local dynamics, while global information is shared across the hierarchical U-Net structure. The model is conditioned on the flow time $t$, which is converted into an embedding and injected at each level of the network, enabling the model to accurately approximate the time-dependent vector field $v_\theta$. The architecture is illustrated in Figure 1.

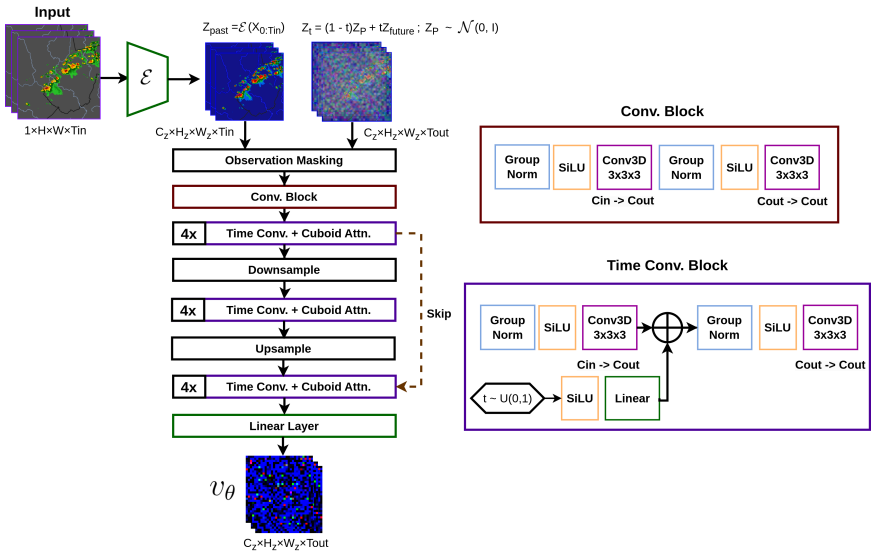


Figure 1: The FlowCast architecture. A U-Net with Cuboid Attention blocks processes latent spatiotemporal data. Conditioning on the flow time $t$ enables the model to learn the time-dependent vector field for generating forecasts.

**Training.** The FlowCast model learns the vector field $v_\theta(Z_t, t, Z_{\text{past}})$. For each data sample:

1. The past ($X_{\text{past}}$) and future ($X_{\text{future}}$) sequences are encoded into latent space ($Z_{\text{past}}$, $Z_{\text{future}}$).
2. A random noise sequence $Z_P \sim \mathcal{N}(0, \mathbf{I})$ and a time step $t \sim U[0, 1]$ are sampled and an interpolated latent state is computed: $Z_t = (1-t)Z_P + tZ_{\text{future}} + \sigma\epsilon$.
3. The model predicts the vector field $\hat{v} = v_\theta(Z_t, t, Z_{\text{past}})$.
4. The training loss is the MSE between predicted and target fields: $\mathcal{L} = \|\hat{v} - (Z_{\text{future}} - Z_P)\|^2$.

**Sampling.** To generate an ensemble of forecasts:

1. The past radar sequence $X_{\text{past}}$ is encoded into its latent representation $Z_{\text{past}}$.
2. For each ensemble member, a distinct initial latent sequence $Z(0)$ is sampled from $\mathcal{N}(0, \mathbf{I})$.
3. The ordinary differential equation (ODE) $\frac{dZ(t)}{dt} = v_\theta(Z(t), t, Z_{\text{past}})$ is solved from $t = 0$ to $t = 1$ with an ODE solver.
4. The resulting forecast at $t = 1$, $Z(1)$, is decoded back into pixel space: $\hat{X}_{\text{future}} = \mathcal{D}(Z(1))$.

The training and sampling procedures are illustrated in Figure 2.

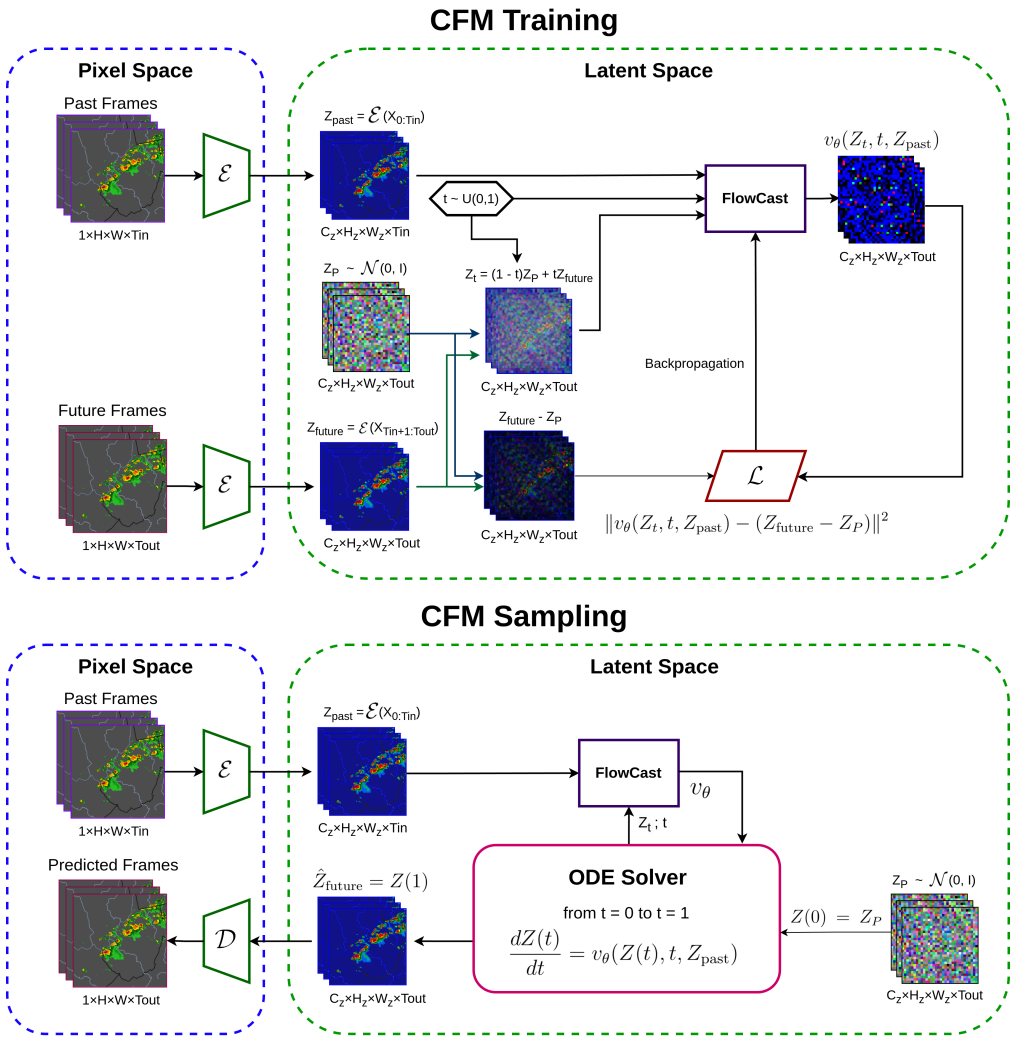


Figure 2: FlowCast training and sampling procedures.

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL SETTING

More details about the experimental setting are provided in Appendix A.1.

### 4.1.1 DATASETS

We evaluate on two 5-minute, 1 km-resolution radar datasets: SEVIR, a US benchmark, and ARSO, a Slovenian composite for a local deployment setting. For both, we predict 12 frames (1 hour) from 13 past frames (65 minutes), per the SEVIR Nowcasting Challenge protocol (Veillette et al., 2020).

Table 1: Summary of datasets used for evaluation.

| Dataset | $N_{train}$ | $N_{val}$ | $N_{test}$ | Resolution | Dimensionality | Interval | Lag/Lead |
|---|---|---|---|---|---|---|---|
| SEVIR | 36,351 | 9,450 | 12,420 | 1 km | 384×384 | 5 min | 13/12 |
| ARSO | 38,229 | 12,743 | 12,744 | 1 km | 301×401 | 5 min | 13/12 |

**SEVIR.** SEVIR (Veillette et al., 2020) provides over 10,000 weather events in a 384×384 km US domain, each spanning 4 hours at 5-minute resolution. We use the 1-km Vertically Integrated Liquid (VIL) field. Following the standard chronological split, we extract 25-frame sequences (13 context, 12 target) with a stride of 12, yielding 36,351 training, 9,450 validation, and 12,420 test samples.

**ARSO.** The ARSO dataset contains 5-minute, 1-km radar reflectivity composites over a 301×401 km Slovenian grid, capturing complex Alpine and coastal dynamics. Using the same 25-frame sequence setup but with stride 1, a 60/20/20 chronological split yields 63,716 training, 12,743 validation, and 12,744 test samples.

### 4.1.2 EVALUATION

**Threshold-based categorical scores:** Following prior work (Veillette et al., 2020; Gao et al., 2023; Gong et al., 2024), we evaluate forecasts by converting radar fields to binary masks at given thresholds and computing the False Alarm Ratio (FAR), Critical Success Index (CSI), and Heidke Skill Score (HSS). We also use a max-pooled CSI over 16x16km regions (CSI-P16-M) to better assess predictions of localized extreme events, crucial for warning systems. We report the mean of these scores across all thresholds ("-M"), and also the CSI at the highest threshold for each dataset.

For SEVIR, we follow the literature in using the thresholds $[16, 74, 133, 160, 181, 219]$. For ARSO, we use the thresholds $[15, 21, 30, 33, 36, 39]$ dBZ, derived through quantile mapping to ensure that each threshold corresponds to approximately the same exceedance probability in both datasets.

**Continuous Ranked Probability Score (CRPS):** We use the CRPS to evaluate probabilistic skill. A lower CRPS indicates a more accurate and sharp forecast. For deterministic forecasts ($N = 1$), CRPS reduces to the Mean Absolute Error.

**Ensemble forecasting:** Let $x_{t,i,j}$ represent the ground truth pixel value at location $(i, j)$ and lead time $t$. All probabilistic models are evaluated using an ensemble of $N = 8$ realizations. For categorical scores, we evaluate the ensemble mean prediction (Metric of Ensemble Mean), first computing the ensemble mean $\hat{x}_{t,i,j} = \frac{1}{N} \sum_{k=1}^{N} \hat{x}_{t,i,j}^{(k)}$ and then the metric on this mean forecast.

### 4.1.3 TRAINING DETAILS

**VAE.** We train a separate Variational Autoencoder (VAE) for each dataset to create a specialized latent space. We follow the architecture and training procedure from Rombach et al. (2022), with a Kullback-Leibler divergence loss weight of 1e-4, the AdamW optimizer with a learning rate of 1e-4, and a batch size of 12. The compressed latent space dimensions are shown in Table 2.

Table 2: VAE latent space dimensions

| Dataset | Original Dimensions ($T_{\text{in}}/T_{\text{out}} \times \mathbf{H} \times \mathbf{W} \times \mathbf{C}_{\text{in}}$) | Latent Dimensions ($T_{\text{in}}/T_{\text{out}} \times \mathbf{H}_z \times \mathbf{W}_z \times \mathbf{C}_z$) |
|---|---|---|
| SEVIR | $13/12 \times 384 \times 384 \times 1$ | $13/12 \times 48 \times 48 \times 4$ |
| ARSO | $13/12 \times 301 \times 401 \times 1$ | $13/12 \times 38 \times 52 \times 4$ |

**FlowCast.** We train our CFM model for 200 epochs using the AdamW optimizer with a learning rate of 5e-4 and a cosine scheduler. We set the standard deviation of the I-CFM probability path to a small constant $\sigma = 0.01$ (Tong et al., 2024). Model checkpoints are maintained using an exponential moving average of weights (Ho et al., 2020), with a decay factor of 0.999, and we keep the model checkpoint with the highest CSI-M evaluated on a subset of the validation set. The model is trained with 4 NVIDIA H100 for 7 days, with a global batch size of 12. Further implementation details are provided in Appendix A.1.

#### 4.1.4 Inference Details

Generating a forecast with FlowCast involves solving the learned ODE to transform a noise-initialized latent sequence into a prediction, conditioned on encoded past observations. Following the procedure outlined in 3.2.2, we use the Euler method (Hairer et al., 1993) with 10 steps as the ODE solver. To generate a probabilistic ensemble forecast, this process is repeated eight times with different initial noise samples $Z(0) \sim \mathcal{N}(0, \mathbf{I})$.

### 4.2 Comparison to the State of the Art

We evaluate FlowCast against three deterministic baselines: U-Net (Veillette et al., 2020), Earthformer (Gao et al., 2022), and SimVPv2 (Tan et al., 2025), as well as two probabilistic baselines: PreDiff (Gao et al., 2023) and CasCast (Gong et al., 2024). All models are trained following their publicly released code, with the training budget fixed at 200 epochs. For probabilistic models, we adopt our evaluation protocol by selecting the checkpoint with the highest CSI-M on a validation subset, using exponential moving average weights to ensure comparability.

As shown in Table 3 and Figure 3, FlowCast sets a new state-of-the-art, achieving the highest overall CSI-M and HSS-M and the lowest CRPS. Its advantage is most pronounced at longer lead times, with a final CSI-219 score 256% higher than the best deterministic model. For detecting localized extreme events (CSI-P16-M), FlowCast is competitive with CasCast, which achieves a 2.8% higher score but with a 17.8% higher False Alarm Ratio (FAR-M). This suggests CasCast's score may stem from over-prediction, while FlowCast better balances detection and precision.

Table 3: Comparison of FlowCast with baseline models on the SEVIR dataset. All metrics are computed over a 12-step forecast, except "Forecast @ +65 min" which only uses the last frame.

| | | Forecast @ 12 steps | | | | Forecast @ +65 min | |
|---|---|---|---|---|---|---|---|
| Model | CRPS ↓ | CSI-M ↑ | CSI-P16-M ↑ | HSS-M ↑ | FAR-M ↓ | CSI-M ↑ | CSI-219 ↑ |
| U-Net | 0.0273 | 0.394 | 0.384 | 0.497 | 0.308 | 0.259 | 0.009 |
| Earthformer | 0.0252 | 0.411 | 0.407 | 0.518 | **0.285** | 0.280 | 0.016 |
| SimVP | 0.0249 | 0.423 | 0.424 | 0.532 | 0.298 | 0.280 | 0.012 |
| PreDiff | 0.0189 | 0.413 | 0.423 | 0.523 | 0.313 | 0.281 | 0.018 |
| CasCast | 0.0201 | 0.442 | **0.520** | 0.562 | 0.383 | 0.311 | 0.054 |
| **FlowCast (ours)** | **0.0182** | **0.460** | 0.506 | **0.580** | 0.325 | **0.324** | **0.057** |

Figure 4 compares forecast sequences from FlowCast with the baselines. FlowCast produces sharp, perceptually realistic forecasts, avoiding the smoothness of deterministic models. More examples are provided in Appendix A.2.1.

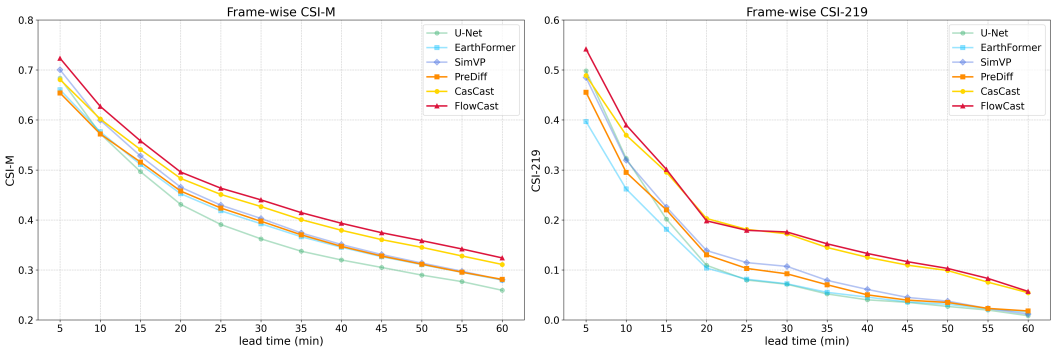


Figure 3: CSI-M and CSI at the 219 threshold per lead time on the SEVIR dataset. FlowCast shows consistent improvement over baselines for CSI-M and avoids the oversmoothing of deterministic models at longer lead times (CSI-219).

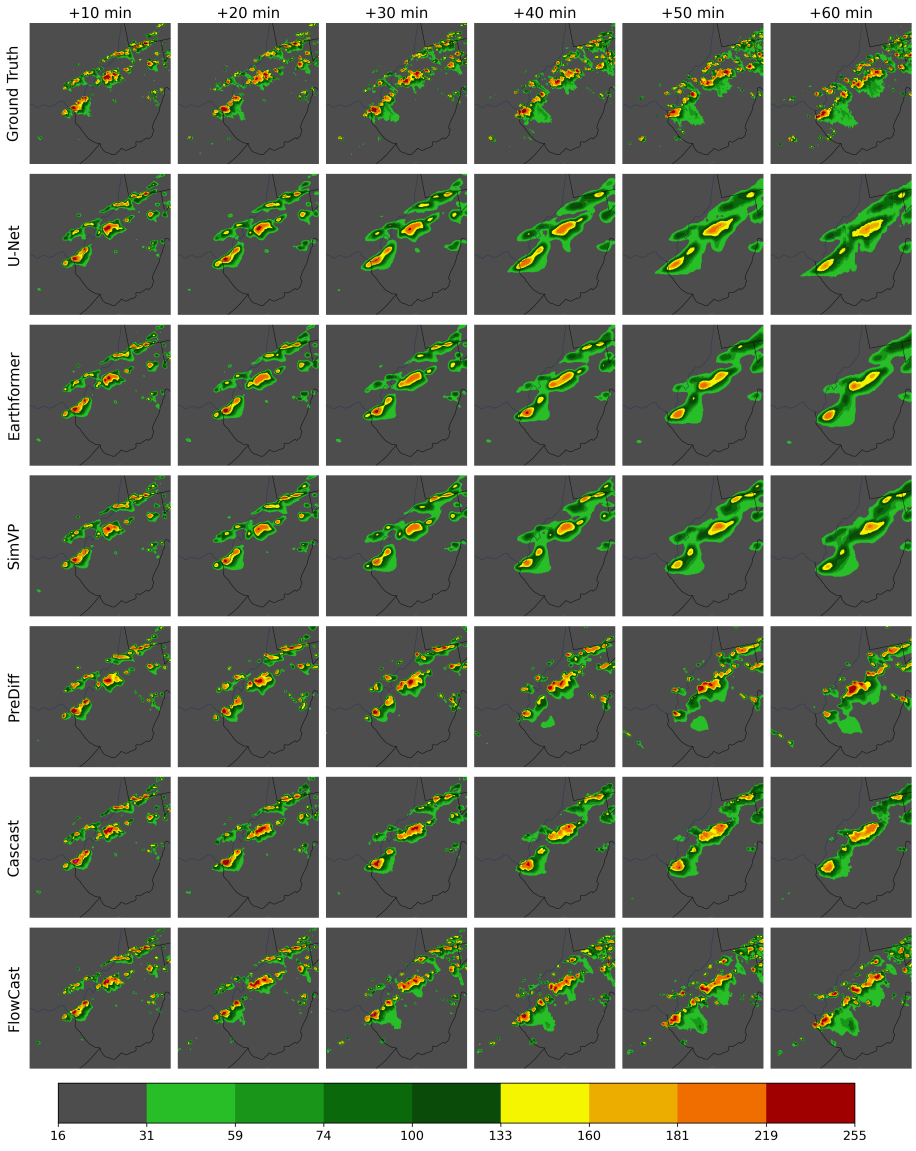


Figure 4: Qualitative comparison of FlowCast with other baselines on a SEVIR sequence. Columns show lead times from 10 to 60 minutes. Rows show the ground-truth, followed by the models.

Similarly, results for the ARSO dataset are shown in Table 4. FlowCast again achieves the best scores in almost all metrics, demonstrating its advantages on a different dataset with distinct local dynamics.

Table 4: Comparison of FlowCast with baseline models on the ARSO dataset. All metrics are computed over a 12-step forecast, except "Forecast @ +65 min" which only considers the last frame.

| | | Forecast @ 12 steps | | | | Forecast @ +65 min | |
|---|---|---|---|---|---|---|---|
| Model | CRPS ↓ | CSI-M ↑ | CSI-P16-M ↑ | HSS-M ↑ | FAR-M ↓ | CSI-M ↑ | CSI-219 ↑ |
| U-Net | 0.0264 | 0.399 | 0.432 | 0.505 | **0.371** | 0.260 | 0.011 |
| Earthformer | 0.0270 | 0.403 | 0.439 | 0.512 | 0.409 | 0.274 | 0.010 |
| SimVP | 0.0267 | 0.415 | 0.462 | 0.526 | 0.401 | 0.288 | 0.029 |
| PreDiff | 0.0211 | 0.369 | 0.411 | 0.471 | 0.400 | 0.241 | 0.010 |
| CasCast | 0.0253 | 0.373 | 0.511 | 0.483 | 0.488 | 0.277 | 0.057 |
| **FlowCast (ours)** | **0.0209** | **0.420** | **0.514** | **0.535** | 0.422 | **0.315** | **0.073** |

### 4.3 Ablation Studies

Due to computational constraints, all ablation studies were run on the first 10% of the SEVIR test set using a single NVIDIA A100 GPU.

#### 4.3.1 CFM objective against Diffusion

To isolate the benefits of the CFM objective, we compare FlowCast against a strong baseline using the same backbone architecture but trained with a diffusion objective. We trained a DDPM (Ho et al., 2020) for 1000 timesteps. For efficient inference, we employed a DDIM sampler (Song et al., 2021) with a varying number of steps. This provides a strong and practical baseline to evaluate FlowCast against a highly optimized diffusion process on the same powerful architecture.

The results in Table 5 clearly demonstrate the superiority of the CFM objective. With just 10 steps, FlowCast (CFM) drastically outperforms the DDIM sampler, even a 100-step DDIM baseline, across key metrics like CRPS and CSI-M.

Table 5: Ablation study: CFM vs. diffusion objective. Results highlight the superior performance and efficiency of the CFM framework. All metrics are computed over a 12-step forecast, except "Forecast @ +65 min" which only considers the last frame.

| | | Forecast @ 12 steps | | | | Forecast @ +65 min | | |
|---|---|---|---|---|---|---|---|---|
| Model | CRPS ↓ | CSI-M ↑ | CSI-P16-M ↑ | HSS-M ↑ | FAR-M ↓ | CSI-M ↑ | CSI-219 ↑ | Time/Seq. (s) |
| CFM (10 steps) | **0.0168** | **0.455** | **0.514** | **0.572** | 0.338 | **0.303** | **0.014** | **24** |
| DDIM (10 steps) | 0.0262 | 0.395 | 0.450 | 0.503 | 0.335 | 0.236 | 0.002 | **24** |
| DDIM (50 steps) | 0.0212 | 0.398 | 0.451 | 0.504 | 0.321 | 0.236 | 0.002 | 120 |
| DDIM (100 steps) | 0.0208 | 0.398 | 0.450 | 0.502 | **0.319** | 0.235 | 0.002 | 239 |

#### 4.3.2 Inference Time Efficiency: Performance vs. Number of Function Evaluations

We assess inference efficiency by comparing FlowCast and the diffusion backbone across a range of function evaluations (NFE), where one NFE is an Euler (CFM) or DDIM step. Each NFE adds 2.4s per 8-member ensemble forecast. Figure 5 shows FlowCast is highly efficient, nearing optimal CRPS and CSI-M scores in just 3-10 steps. In contrast, the diffusion model requires 20-50 steps to peak and degrades sharply below 10 NFE. These results highlight the superior efficiency of the CFM framework, which learns a more direct mapping to the data manifold and enables high-fidelity forecasts with significantly fewer model evaluations. This efficiency is a crucial advantage for operational settings where forecasts must be both rapid and reliable.

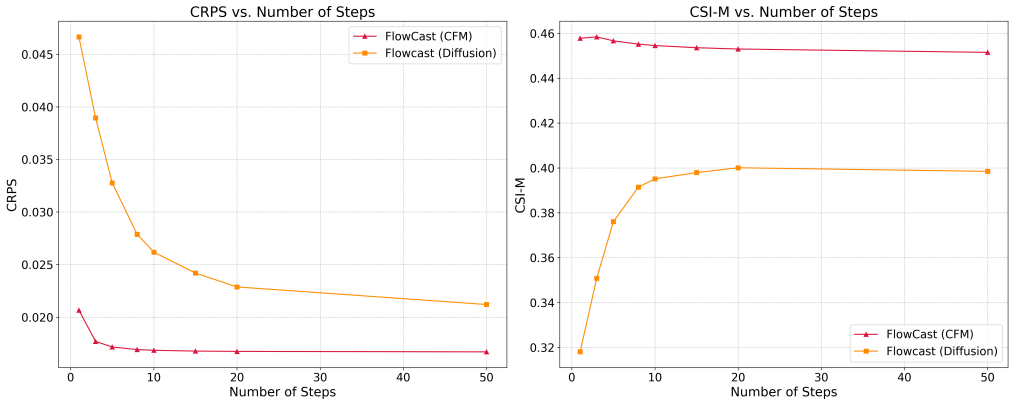


Figure 5: Performance vs. efficiency trade-off. Forecast quality (CRPS ↓, CSI-M ↑) as a function of the Number of Function Evaluations (NFE). FlowCast (CFM) achieves near-optimal performance with only 3 to 10 steps, while the DDIM-based model requires 20 steps for CSI-M and 50 steps for CRPS, and degrades sharply at low NFE.

## 5 Conclusion

In this paper, we introduced FlowCast, the first model to apply Conditional Flow Matching (CFM) to precipitation nowcasting. Our experiments on the SEVIR and ARSO datasets show that FlowCast achieves state-of-the-art performance. Through direct ablation studies, we showed that the CFM objective is not only more accurate than a traditional diffusion objective on the same architecture but also vastly more efficient. FlowCast maintains high forecast quality with as few as a single sampling step, a regime where diffusion models fail. Our results firmly establish CFM as a powerful, efficient, and practical alternative for high-dimensional spatiotemporal forecasting.

**Limitations and Future Work:** While FlowCast shows significant promise, we identify two primary areas for future development. First, its reliance solely on radar data could be a limitation. Future work should explore multi-modal data fusion (e.g., satellite, NWP) to enhance robustness and accuracy. Second, our evaluation was limited to two datasets due to computational cost; a broader study across more meteorological regimes is needed to confirm generalizability.

### Reproducibility Statement

To ensure the reproducibility of our work, we have provided comprehensive supporting materials. **Code:** The full source code for our FlowCast model, including scripts for training and evaluation, is included in the supplementary material. The repository contains detailed instructions for setting up the required software environment and running the experiments. **Datasets:** Our work utilizes two datasets. The SEVIR dataset is a public benchmark, and details for access are provided by Veillette et al. (2020). The ARSO dataset was provided by the Slovenian Environment Agency (ARSO) for this research; we are actively collaborating with the agency to facilitate its public release in the near future. **Experimental Details:** Section 4.1 of the main paper provides a detailed description of our experimental setup. Further implementation details are available in Appendix A.1, including all model hyperparameters.

## References

Sojung An, Tae-Jin Oh, Eunha Sohn, and Donghyun Kim. Deep learning for precipitation nowcasting: A survey from the perspective of time series forecasting. *Expert Systems with Applications*, 268:126301, 2025.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

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

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang (Bernie) Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 25390–25403. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/a2affd71d15e8fedffe18d0219f4837a-Paper-Conference.pdf.

Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle Maddix, Yi Zhu, Mu Li, and Yuyang (Bernie) Wang. Prediff: Precipitation nowcasting with latent diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 78621–78656. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f82ba6a6b981fbbecf5f2ee5de7db39c-Paper-Conference.pdf.

Urs Germann and Isztar Zawadzki. Scale-dependence of the predictability of precipitation from continental radar images. part i: Description of the methodology. *Monthly Weather Review*, 130 (12):2859–2873, 2002. doi: 10.1175/1520-0493(2002)130⟨2859:SDOTPO⟩2.0.CO;2.

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

E Hairer, Syvert P Norsett, and Gerhard Wanner. *Solving ordinary differential equations I: Nonstiff problems*. Springer, Berlin, Germany, 2 edition, 1993.

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

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal Flow Matching for Efficient Video Generative Modeling. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YOUR_PAPER_ID_HERE.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Jussi Leinonen, Ulrich Hamann, and Urs Germann. Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. In *EGU General Assembly*, 2023. doi: 10.5194/egusphere-egu23-9531.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Rachel Prudden, Samantha V. Adams, Dmitry Kangin, Niall H. Robinson, Suman V. Ravuri, Shakir Mohamed, and Alberto Arribas. A review of radar-based nowcasting of precipitation and applicable machine learning techniques. *CoRR*, abs/2005.04988, 2020. URL https://arxiv.org/abs/2005.04988.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021. doi: 10.1038/s41586-021-03854-z.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Cheng Tan, Zhangyang Gao, Siyuan Li, and Stan Z Li. Simvpv2: Towards simple yet powerful spatiotemporal predictive learning. *IEEE Transactions on Multimedia*, 2025.

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. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Mark Veillette, Siddharth Samsi, and Chris Mattioli. Sevir: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in Neural Information Processing Systems*, 33:22009–22019, 2020.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017.

Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S. Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, 2023. doi: 10.1109/TPAMI.2022.3165153.

Hao Wu, Yuxuan Liang, Wei Xiong, Zhengyang Zhou, Wei Huang, Shilong Wang, and Kun Wang. Earthfarsser: Versatile spatio-temporal dynamical systems modeling in one model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15906–15914, 2024.

# A Appendix

## A.1 Implementation Details

This section details the implementation of FlowCast. The code to train and evaluate FlowCast is provided as supplementary material.

### A.1.1 VAE

The VAE was configured with the following hyperparameters:

Table 6: VAE hyperparameter summary

| Category | Parameter | Value / Setting |
|---|---|---|
| ***Dataset*** | | |
| | Source | SEVIR (`vil`) & ARSO (`zm`) |
| | Input Dimensionality (per frame) | $384 \times 384 \times 1$ (SEVIR), $301 \times 401 \times 1$ (ARSO) |
| | Input Preprocessing | Frame values scaled to $[0, 1]$ |
| ***Training Objective*** | | |
| | Loss Components | Reconstruction + KL Divergence + Adversarial |
| | KL Divergence Weight ($\lambda_{KL}$) | $1 \times 10^{-4}$ |
| | Discriminator Weight ($\lambda_{adv}$) | 0.5 |
| | Adversarial Loss Type | Hinge Loss |
| | Discriminator Architecture | PatchGAN (Isola et al., 2017) |
| | Discriminator Activation Warmup | 35 epochs (SEVIR), 15 epochs (ARSO) |
| ***Optimization*** | | |
| | Optimizer (Generator & Disc.) | AdamW |
| | Learning Rate (Initial) | $1 \times 10^{-4}$ |
| | Weight Decay | $1 \times 10^{-5}$ |
| | AdamW Betas | $(0.9, 0.999)$ |
| | LR Scheduler | Cosine Annealing with Linear Warmup |
| | LR Warmup Fraction | 20% of total training steps |
| | LR Min Warmup Ratio | 0.1 |
| | Min. LR Ratio | $10^{-3}$ |
| ***Training Configuration*** | | |
| | Batch Size | 12 (Global), 3 (Local) |
| | Max. Number of Epochs | 250 |
| | Gradient Clipping Norm | 1.0 |
| | Early Stopping Patience | 50 epochs |
| | Early Stopping Metric | Generator validation loss |
| | Training Nodes | $4 \times$ H100 GPUs |
| | FP16 Training | Disabled |
| ***Model Configuration*** | | |
| | Latent Channels | 4 |
| | GroupNorm Num | 32 |
| | Layers per Block | 2 |
| | Activation Function | SiLU |
| | Encoder-Decoder Depth | 4 |
| | Block Out Channels | [128, 256, 512, 512] |

### A.1.2 FlowCast

**Architecture.** The FlowCast architecture, adapted from Earthformer-UNet (Gao et al., 2023) for latent-space Conditional Flow Matching, has the following configuration:

- **Core U-Net Architecture:**
  - **Hierarchical Stages:** A U-Net with 2 hierarchical stages (one level of downsampling/upsampling within the main U-Net body, in addition to initial/final processing).
  - **Stacked Cuboid Self-Attention Modules:** Each stage in both the contracting (encoder) and expansive (decoder) paths contains a depth of 4 Stacked Cuboid Self-Attention modules.
  - **Base Feature Dimensionality:** 196 units.
- **Spatial Processing:**
  - **Downsampling:** Achieved using Patch Merge (reducing spatial dimensions by a factor of 2 and doubling channel depth).
  - **Upsampling:** Uses nearest-neighbor interpolation followed by a convolution (halving channel depth).
- **Cuboid Self-Attention Details:**
  - **Pattern:** Follows an axial pattern, processing temporal, height, and width dimensions sequentially.
  - **Attention Heads:** 4 attention heads.
  - **Positional Embeddings:** Relative positional embeddings are used.
  - **Projection Layer:** A final projection layer is part of the attention block.
  - **Dropout Rates:** Dropout rates for attention, projection, and Feed-Forward Network (FFN) layers are set to 0.1.
- **Global Vectors:** The specialized global vector mechanism from the original Earthformer is disabled.
- **FFN and Normalization:**
  - **FFN Activation:** Feed-forward networks within the attention blocks use GELU activation.
  - **Normalization:** Layer normalization is applied throughout the relevant parts of the network.
- **Embeddings:**
  - **Spatiotemporal Positional Embeddings:** Added to the input features after an initial projection.
  - **CFM Time Embeddings ($t$):**
    - **Generation:** Generated with a channel multiplier of 4 relative to the base feature dimensionality (resulting in $196 \times 4 = 784$ embedding channels).
    - **Incorporation:** Injected into the network at each U-Net stage using residual blocks that fuse the time embedding with the feature maps (`TimeEmbedResBlock` modules).
- **Skip Connections:** Standard U-Net additive skip connections merge features from the contracting path to the expansive path.
- **Padding:** Zero-padding is used where necessary to maintain tensor dimensions during convolutions or cuboid operations.

**Training and Inference Hyperparameters.** The FlowCast training and inference hyperparameters are as follows:

Table 7: FlowCast hyperparameter summary

| Category | Parameter | Value / Setting |
|---|---|---|
| ***Dataset*** | | |
| | Source | Latent-space sequences from VAE |
| | Input Dimensionality | $13 \times 48 \times 48 \times 4$ (SEVIR), $13 \times 38 \times 52 \times 4$ (ARSO) |
| | Input Preprocessing | Standardized with training set statistics (mean, std) |
| ***Training Configuration*** | | |
| | Loss Function | MSE: $\mathcal{L} = \|\hat{v} - (Z_{\text{future}} - Z_P)\|^2$ |
| | Batch Size | 12 (Global), 3 (Local) |
| | Max. Number of Epochs | 200 |
| | Gradient Clipping Norm | 1.0 |
| | Early Stopping Patience | 50 epochs |
| | Early Stopping Metric | CSI-M evaluated on subset (40 batches) of validation set |
| | Training Nodes | $4 \times$ H100 GPUs |
| | FP16 Training | Enabled |
| | Exponential Moving Average Weights | Enabled |
| | Exponential Moving Average Weights Decay | 0.999 |
| ***Optimization*** | | |
| | Optimizer | AdamW |
| | Learning Rate (Initial) | $5 \times 10^{-4}$ |
| | Weight Decay | $1 \times 10^{-4}$ |
| | AdamW Betas | $(0.9, 0.999)$ |
| | LR Scheduler | Cosine Annealing with Linear Warmup |
| | LR Warmup Fraction | 1% of total training steps |
| | LR Min Warmup Ratio | 0.1 |
| | Min. LR Ratio | $10^{-2}$ |
| ***CFM Parameters*** | | |
| | $\sigma$ | 0.01 |
| | ODE Solver | Euler Method with 10 steps |

**Choice of ODE Solver.** We conducted an ablation study to compare various ODE solvers, including adaptive methods (Adaptive Heun, Dormand-Prince 5) and fixed-step methods (Euler, Midpoint, Runge-Kutta 4). Since no significant performance differences were observed, we selected the Euler method with 10 steps for its computational efficiency and simplicity.

### A.1.3 Evaluation Metrics

**Continuous Ranked Probability Score (CRPS).** The CRPS is evaluated directly at the original data resolution, without applying any spatial pooling. For each ensemble of $N$ forecast members, CRPS is calculated at every pixel and then averaged across all spatial positions and forecast lead times to obtain a single summary metric. If the predictive distribution $F$ at a given pixel and time step is approximated by a Gaussian with mean $\mu$ and standard deviation $\sigma$ (estimated from the ensemble), and $x$ is the observed value, the CRPS can be computed as:

$$\text{CRPS}(F, x) = \sigma \left( \frac{x - \mu}{\sigma} \left( 2\Phi \left( \frac{x - \mu}{\sigma} \right) - 1 \right) + 2\phi \left( \frac{x - \mu}{\sigma} \right) - \frac{1}{\sqrt{\pi}} \right), \tag{1}$$

where $\Phi$ and $\phi$ denote the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution, respectively.

**Threshold-based categorical metrics.** For each chosen intensity threshold $u$, we binarize the continuous ground truth field $x_{t,i,j}$ to obtain an observation mask $\mathbb{1}[x_{t,i,j} > u]$. The representative forecast $\hat{x}_{t,i,j}$ (see Section 4.1.2) is likewise thresholded to produce a binary forecast mask $\mathbb{1}[\hat{x}_{t,i,j} > u]$.

Using these binary masks and the dataset-specific thresholds, we construct a $2 \times 2$ contingency table for each evaluation:

| | Observ. yes | Observ. no |
|---|---|---|
| Forecast yes | $H$ | $F$ |
| Forecast no | $M$ | $C$ |

$H$ : Hits (True Positives),
$M$ : Misses (False Negatives),
$F$ : False Alarms (False Positives),
$C$ : Correct Negatives.

The values $H, M, F, C$ are summed over all spatial locations, batches, and forecast lead times. The following metrics are then computed:

**False Alarm Ratio (FAR):** The proportion of forecasted events that did not actually occur.

$$\mathrm{FAR} = \frac{F}{H + F}. \tag{2}$$

**Critical Success Index (CSI):** The fraction of observed and/or forecasted events that were correctly predicted, ignoring correct negatives. This metric is sensitive to both missed events and false alarms.

$$\mathrm{CSI} = \frac{H}{H + M + F}. \tag{3}$$

**Heidke Skill Score (HSS):** The accuracy of the forecast relative to random chance.

$$\mathrm{HSS} = \frac{2(HC - MF)}{(H + M)(M + C) + (H + F)(F + C)}. \tag{4}$$

**Pooled CSI (CSI-P16):** Forecasts and ground truth fields are first downsampled by applying a max-pooling operation over non-overlapping $16 \times 16$ pixel blocks. CSI is then recomputed on these pooled fields. A hit anywhere within a $16 \times 16$ block is registered as a success, thereby rewarding models that capture the local presence and intensity of precipitation, even if exact pixel-level alignment is imperfect.

### A.2 More Qualitative Examples

This section presents additional qualitative examples comparing FlowCast against the baseline models

### A.2.1 SEVIR


Figure 6: Qualitative comparison of FlowCast with other baselines on a SEVIR sequence.

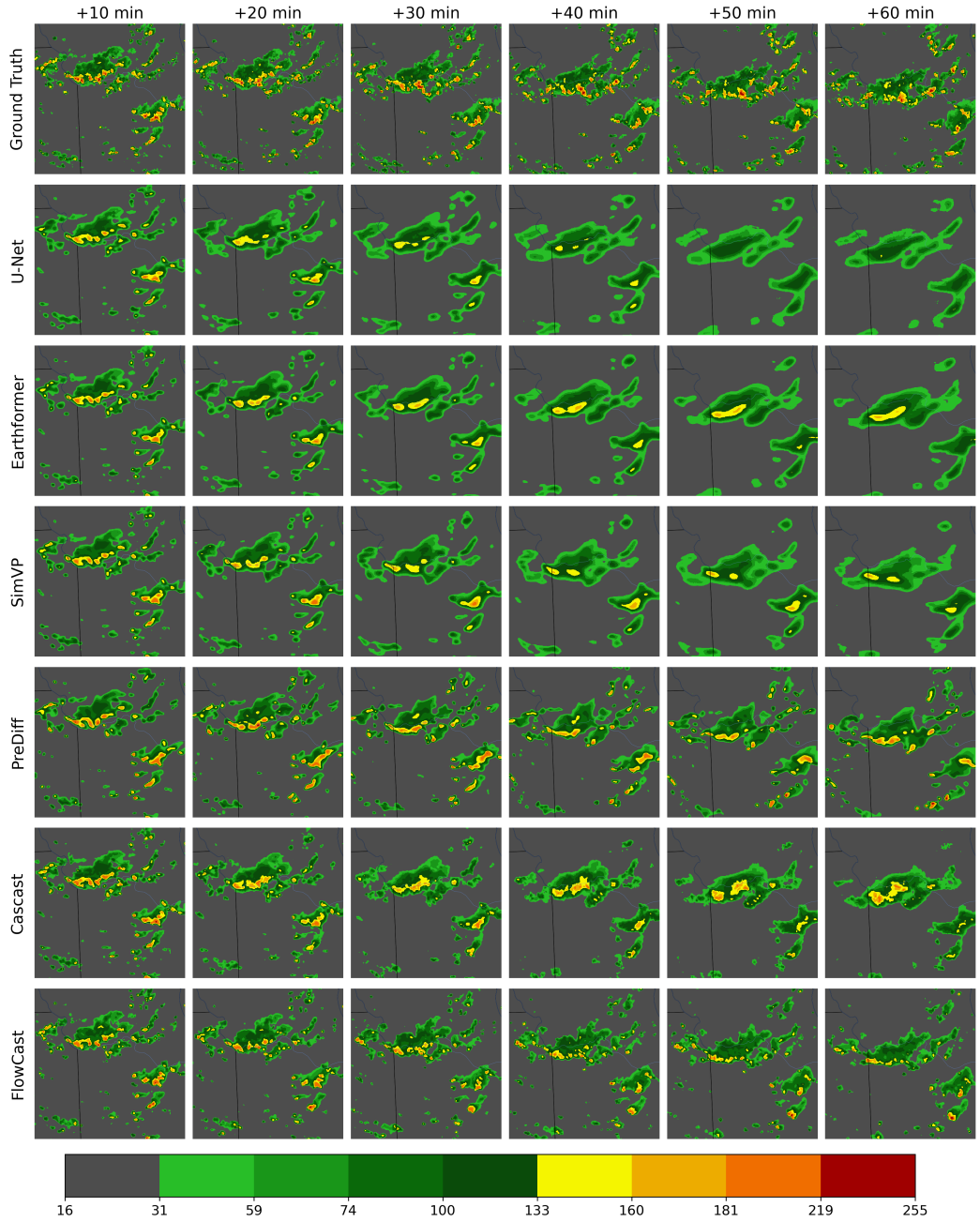


Figure 7: Qualitative comparison of FlowCast with other baselines on a SEVIR sequence.

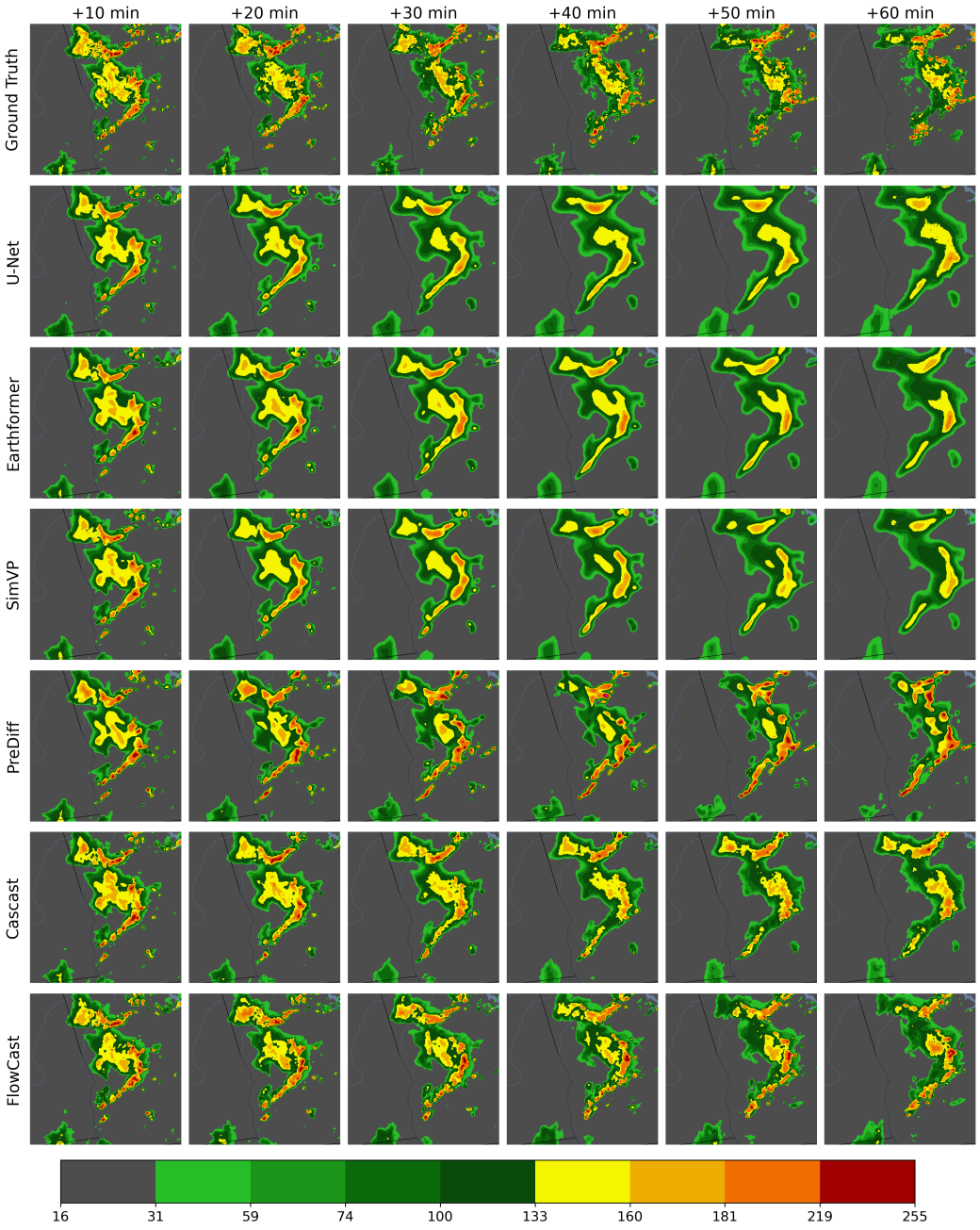


Figure 8: Qualitative comparison of FlowCast with other baselines on a SEVIR sequence.

### A.2.2 ARSO

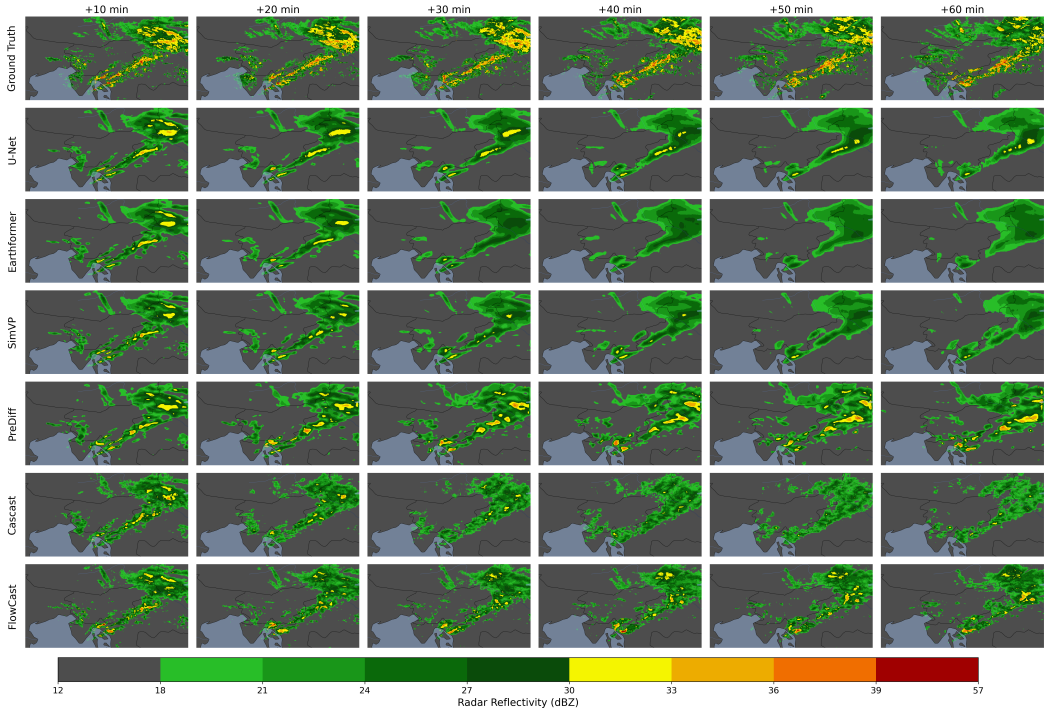


Figure 9: Qualitative comparison of FlowCast with other baselines on an ARSO sequence.

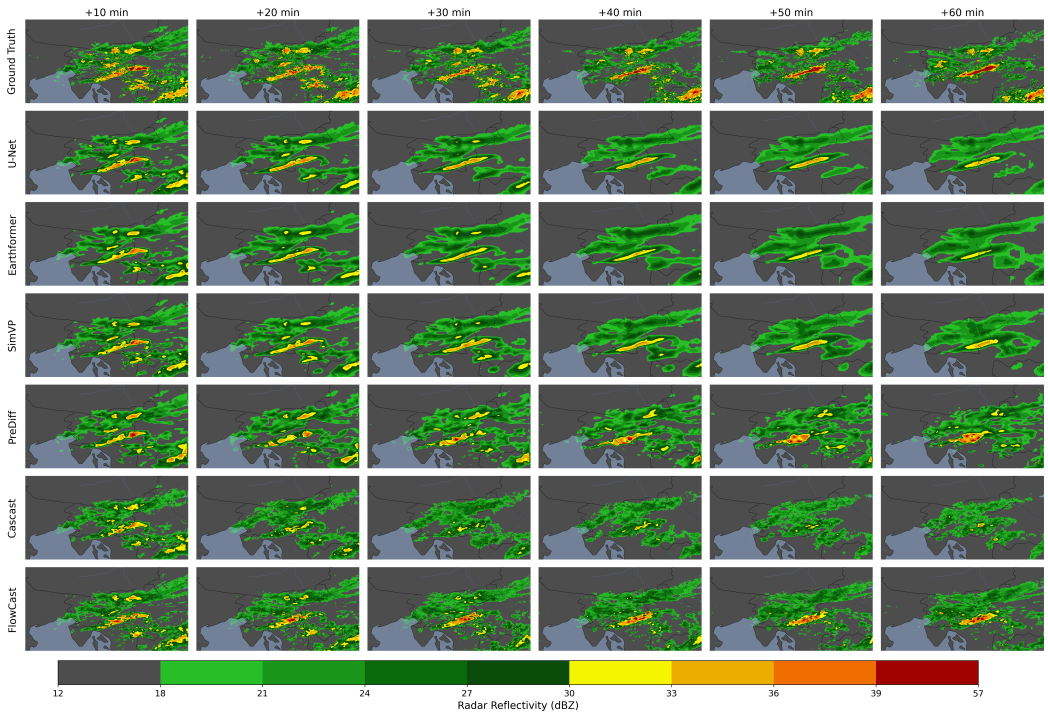


Figure 10: Qualitative comparison of FlowCast with other baselines on an ARSO sequence.

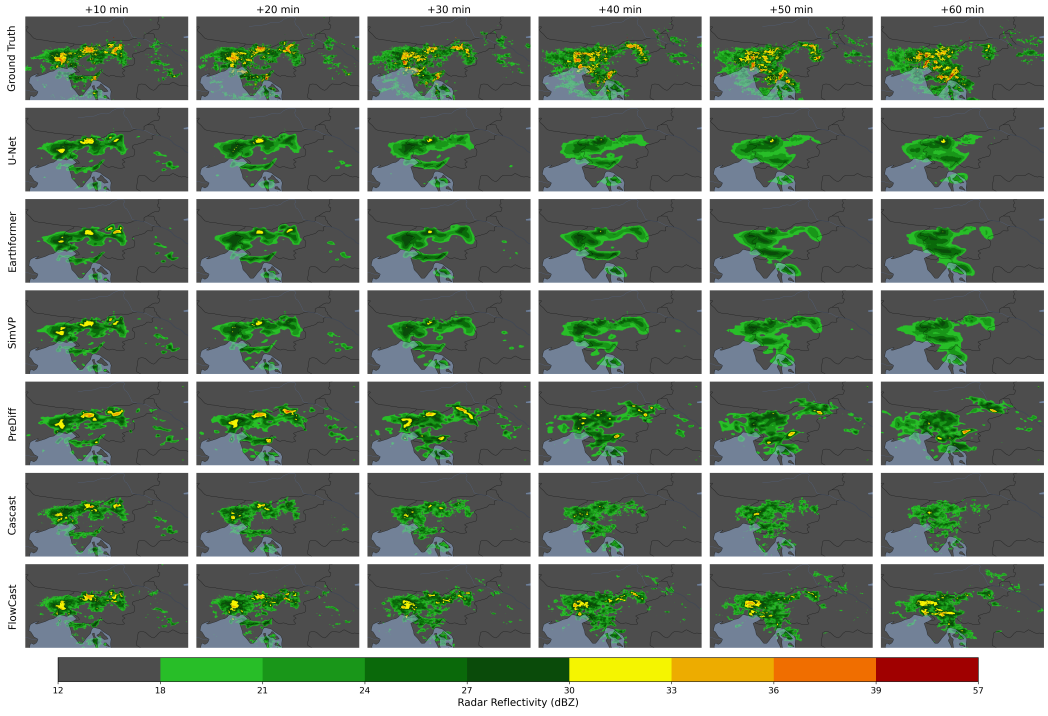


Figure 11: Qualitative comparison of FlowCast with other baselines on an ARSO sequence.

### A.3 LLM USAGE STATEMENT

During the preparation of this manuscript, we utilized a large language model (LLM) as a writing assistant. The LLM's primary role was to help refine sentence structure, improve clarity, and ensure grammatical correctness and consistency in tone. All scientific contributions, including the research ideation, methodological design, experimental analysis, and interpretation of results, were conducted solely by the authors, who take full responsibility for the content of this work.