JUST WEATHER SCORING: EFFICIENT END-TO-END NOWCASTING WITH DISTRIBUTIONAL DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative diffusion models are well-suited for probabilistic precipitation nowcasting, but existing approaches often rely on separately trained compression or deterministic forecasting components and remain costly at inference due to iterative denoising. We introduce *Just Weather Scoring* (JWS), a single-stage, end-to-end diffusion model which addresses both issues by forecasting directly in radar space and enabling few-step generation. Radar-space modeling greatly simplifies training and inference and eliminates uncertainty arising from lossy compression. JWS combines *Masked Asynchronous Diffusion*, a timestep-sampling scheme that preserves clean context while adapting diffusion training to high-dimensional spatio-temporal data, with a simple scoring-rule objective that aligns training with probabilistic forecasting and unlocks few-step generation. On the SEVIR and MeteoNet benchmarks, JWS achieves state-of-the-art probabilistic forecasting performance at reduced training and inference cost. Even our smallest model remains competitive using substantially fewer parameters and more than $17\times$ faster inference. Code and models will be released upon acceptance.

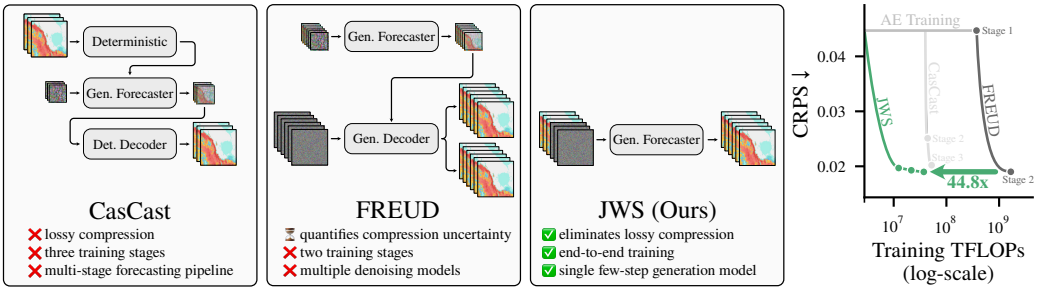


Figure 1: **Nowcasting Made Easy.** JWS, an end-to-end, pixel-space nowcasting model, simplifies training and inference while $44\times$ more efficient to train than the state-of-the-art FREUD model with comparable CRPS. Beyond simplification, end-to-end inference eliminates compression uncertainty.

1 INTRODUCTION

Short-term, high-resolution precipitation nowcasting is essential for applications ranging from renewable energy planning (Sperati et al., 2016; Wu et al., 2018) to emergency response (Salio et al., 2025). Useful nowcasts must resolve spatial details, update rapidly, and quantify uncertainty arising from chaotic dynamics and limited observability (Ravuri et al., 2021). Compared to numerical weather prediction, machine learning offers faster inference and learns directly from data. Yet, regression averages plausible forecasts and thus produces increasingly blurry forecasts as uncertainty grows. Generative models instead sample plausible precipitation from the conditional distribution, enabling sharp individual forecasts and ensemble-based uncertainty quantification. Reliable probabilistic forecasting requires representing the full range of possibilities rather than concentrating on a few outcomes. Therefore, diffusion models are a natural fit due to their distributional coverage.

However, many diffusion-based nowcasting approaches rely on multi-stage training pipelines. Inspired by image generation (Rombach et al., 2022), latent space methods first train a compression

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.

model and learn denoising of forecasts only on compressed representations to reduce cost (Gao et al., 2023; Leinonen et al., 2023). Other approaches use a deterministic forecast to condition generation (Gong et al., 2024) or define a residual prediction task (Yu et al., 2024), resulting in up to three components. Beyond training complexity, these components introduce uncertainty and biases. While visually minor imperfections are acceptable in image synthesis, small errors in precipitation intensity and location can have a consequential impact. FREUD (Schusterbauer et al., 2026b) partially addresses this concern by modeling uncertainty from lossy compression with a generative decoder. Nevertheless, it retains a separate compression stage, and quantification of compression uncertainty requires ensembling many reconstructions per forecast, leading to multiplicative cost.

Therefore, prior work leaves the central challenge unresolved: how to build a simple, uncertainty-aware nowcasting model that makes efficient use of training and inference compute. We introduce *Just Weather Scoring* (JWS), an end-to-end generative model that avoids additional stages and operates directly on uncompressed data. We enable forecasting with a single transformer that is parametrized to predict clean radar maps (Li & He, 2025) and downsamples only by patchification. Our single-stage approach is more efficient and capable than pipelines with multiple trained components. Further, we introduce *Masked Asynchronous Diffusion* (MAD), preserving clean conditioning while sampling correlated, framewise noise levels for predicted timesteps, aligning training with forecasting. Thus, by training with MAD, JWS supports flexible conditioning. Spatiotemporal resolution-aware timestep sampling in MAD adapts corruption levels to uncompressed radar data.

As iterative denoising remains costly, we aim to reduce sampling steps. Flow Map methods (Boffi et al., 2025a) reduce denoising steps by directly learning the integration over diffusion time. However, canonical approaches (Geng et al., 2025; Frans et al., 2025; Boffi et al., 2025b) incur substantial overhead to construct training targets. Functional Generative Networks (FGNs) (Alet et al., 2025) use scoring-rule supervision to directly optimize probabilistic forecasts but introduce stochasticity only by noise injection into deterministic predictors, potentially limiting distributional coverage.

We seek to combine scoring-rule supervision with diffusion model coverage. Distributional Diffusion Models (DDM) (De Bortoli et al., 2025) replace vector field regression with scoring-rule supervision, training denoising models to represent the conditional posterior through ensembles of vector fields. With a full conditional posterior, reverse transitions recover the data distribution even on a coarse time grid, enabling few-step inference (De Bortoli et al., 2025). However, balancing few- and many-step generation previously required careful tuning of interdependent variables. We propose *simple Distributional Diffusion Models* (sDDM), which are trained with a hyperparameter-free objective, linearly interpolating distributional CRPS and squared error.

With MAD and sDDM, JWS generates radar sequences directly in pixel space, eliminating additional pipeline stages and allocating computation entirely to generative forecasting. Our key insight is that standard transformers can support efficient end-to-end nowcasting by combining task-specific timestep sampling with distributional supervision. This yields simple and efficient training and inference with superior probabilistic forecasting, without lossy compression or deterministic priors.

Our contributions are as follows:

- **End-to-end training:** JWS is trained directly on sensor data without separately trained compression or deterministic forecasting stages, simplifying training and inference.
- **Nowcasting-centric Recipe:** *Masked Asynchronous Diffusion* aligns conditioning observed during training and inference while preserving test-time flexibility.
- **Simple distributional objective:** *simple Distributional Diffusion Models* are trained with a hyperparameter-free objective, improving forecasting and enabling few-step sampling.
- **Efficient Nowcasting:** Taken together, these components enable even our smallest model to achieve competitive nowcasting with approximately $17\times$ fewer parameters than FlowCast and substantially lower inference cost than prior methods.

2 Related Work

An extended Related Work section is found in Appendix Sec. D.

Precipitation Nowcasting. Nowcasting refers to short-term, high-resolution forecasting. As rapid updates are typically required, numerical weather prediction systems (NWP) are unsuitable for precipitation nowcasting (Shi et al., 2015; Leinonen et al., 2023; Ravuri et al., 2021). Machine learning offers faster inference (Shi et al., 2015; Agrawal et al., 2019; Wang et al., 2017; Gao et al., 2022b). Yet, training with regression objectives yields blurry forecasts under uncertainty, thus the field shifted towards generative nowcasting methods (Ravuri et al., 2021; Zhang et al., 2023; Leinonen et al., 2023; Gao et al., 2023). Initially, generative adversarial networks were used (Goodfellow et al., 2014; Ravuri et al., 2021; Zhang et al., 2023; Ji et al., 2022), yet adversarial training can induce mode collapse (Kossale et al., 2022). Therefore, diffusion-based nowcasting has become popular due to strong mode coverage and a principled mathematical foundation (Ho et al., 2020; Dhariwal & Nichol, 2021; Lipman et al., 2023). To reduce denoising complexity, two-stage latent diffusion methods are commonly used (Leinonen et al., 2023; Gao et al., 2023; Gong et al., 2024; Ribeiro & Pucer, 2026; Schusterbauer et al., 2026b) or generation is conditioned on deterministic priors (Yu et al., 2024; Gong et al., 2024; Zhu et al., 2026). Yet, FREUD (Schusterbauer et al., 2026b) finds these priors reduce distributional coverage, and inherently lossy compression induces additional uncertainty. They ensemble generative reconstructions, introducing computational overhead and maintaining separately trained stages. Alet et al. (2025) propose to directly supervise ensembles and predict multiple forecasts by perturbing parameters through noise injection. Yet, all stochasticity must be captured by the noise vector, thereby potentially limiting coverage. We instead propose to train end-to-end radar-space nowcasting, eliminating compression stages completely while retaining diffusion model coverage.

Pixel-space Modeling. Latent diffusion (Rombach et al., 2022) enables high-resolution image (Esser et al., 2024) and video (Ho et al., 2022) generation. Perceptual (Zhang et al., 2018) and adversarial (Goodfellow et al., 2014) losses typically encourage losing mostly imperceptible details. However, the importance of small-scale variations in nowcasting limits applicability. Previously, architecture adaptations for pixel-space generation were explored (Hoogeboom et al., 2023; Yu et al., 2025; Crowson et al., 2024). Recently, Li & He (2025) showed that standard transformers are indeed suitable. However, pixel-space nowcasting is especially challenging due to high data dimensionality. While RainDiff (Nguyen et al., 2025b) and PixelFlowCast (Zhu et al., 2026) perform pixel-space nowcasting, they rely on specialized architectures or only model residuals of deterministic forecasts. In comparison, we use standard transformers for end-to-end precipitation nowcasting.

Few-Step Sampling. Iterative denoising remains a bottleneck of diffusion. Distillation (Salimans & Ho, 2022; Sauer et al., 2023; Song et al., 2023; Yin et al., 2024b) drastically reduces latency, but requires two-stage training. Flow Maps (Boffi et al., 2025a) directly integrate larger steps. However, prominent MeanFlows (Geng et al., 2025) induce substantial training overhead, while Shortcut models (Frans et al., 2025) invest compute to produce self-bootstrapped targets. Alternatively, Distributional Diffusion Models (DDM) (De Bortoli et al., 2025) train denoising models with ensembles and scoring rule supervision, enabling few-step inference. Moreover, distributional supervision aligns with probabilistic forecasting (Alet et al., 2025). To scale DDM training, improved Distributional Diffusion Models (iDDM) (Anonymous, 2026) expand the ensemble only for final layers. Yet, their loss relies on tuning interdependent parameters. We replace iDDM's tuned parameter schedules with a simple interpolation between CRPS and squared error, achieving better nowcasting results without additional loss hyperparameters.

3 METHOD

3.1 PRELIMINARIES

Precipitation nowcasting. Given C radar observations $\mathbf{x}^{1:C}$, with $\mathbf{x}^t \in \mathbb{R}^{H \times W}$ at time t, the goal of probabilistic precipitation nowcasting is to model the distribution $p(\mathbf{x}^{C+1:C+L} \mid \mathbf{x}^{1:C})$ of the next L precipitation frames. At inference, we sample multiple sequences to form a forecast ensemble, keeping the observed past fixed.

Spatiotemporal diffusion. Diffusion models learn to transform samples from a simple prior distribution into samples from the data distribution through denoising (Ho et al., 2020; Albergo et al., 2023; Lipman et al., 2023). In our setting, each sample is a complete radar sequence $\mathbf{x}_1 \in \mathbb{R}^{T \times H \times W}$

of length $T = C + L$. Given a clean sequence $\mathbf{x}_1 \sim p_{\text{data}}$ and independent Gaussian noise $\mathbf{x}_0 \sim \mathcal{N}(0, \mathbf{I})$, we use a linear interpolant to construct noisy sequences at diffusion time $\tau \in [0, 1]$:

$$\mathbf{x}_\tau = \tau \mathbf{x}_1 + (1 - \tau)\mathbf{x}_0, \qquad \text{with} \quad \tau \sim p(\tau). \tag{1}$$

Here $\tau \in [0, 1]$ controls corruption from noise ($\tau = 0$) to clean data ($\tau = 1$) and differs from the frame index t. The linear path has velocity $\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_0$. A Flow Matching (Lipman et al., 2023) model $\mathbf{v}_\theta$ is trained to predict this velocity by minimizing the squared error

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, \tau}\left[\|\mathbf{v} - \mathbf{v}_\theta(\mathbf{x}_\tau, \tau)\|_2^2\right]. \tag{2}$$

At inference, we initialize a Gaussian noise sequence and numerically integrate the learned ODE $d\mathbf{x}_\tau / d\tau = \mathbf{v}_\theta(\mathbf{x}_\tau, \tau)$ from $\tau = 0$ to $\tau = 1$. The resulting sample follows the model distribution p_θ, approximating p_{data} (Lipman et al., 2023). Sampling independent conditional forecasts enables ensemble-based uncertainty estimation. As ensemble size increases, sample variance converges to the learned predictive variance, which approximates aleatoric uncertainty (Chan et al., 2024).

Standard sequence diffusion uses a single diffusion time shared across frames. Hence, it corrupts all frames to the same degree. More generally, framewise corruption can be described by $\boldsymbol{\tau} = (\tau^1, \ldots, \tau^T)$, with $\mathbf{x}_{\boldsymbol{\tau}}^t = \tau^t \mathbf{x}_1^t + (1 - \tau^t)\mathbf{x}_0^t$. This accommodates clean context ($\tau^t = 1$) alongside noisy frames ($\tau^t < 1$), whose diffusion times may be shared or vary across the sequence.

3.2 Pixel-Space Nowcasting

We aim to perform spatiotemporal diffusion directly on sensor data, generating radar fields in a single stage (Fig. 1). Eliminating latent compression avoids the associated uncertainty from lossy compression, devoting the inference budget to ensembling forecasts (Fig. 2). To support high-dimensional diffusion, we adapt conditioning, prediction targets, and training timestep sampling.

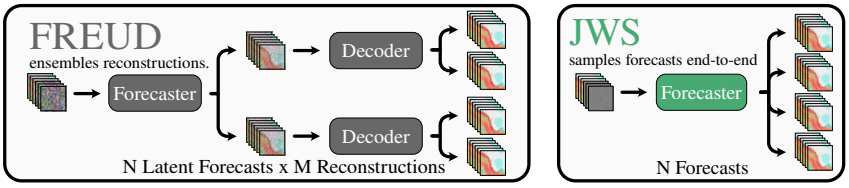


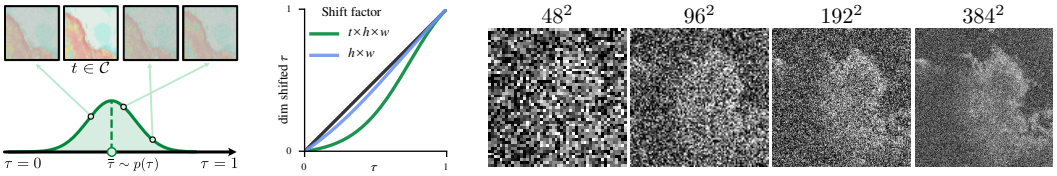
Figure 2: **Forecast ensembles in pixel space. Left:** FREUD (Schusterbauer et al., 2026b) samples multiple reconstructions per forecast to represent reconstruction uncertainty. **Right:** JWS directly samples radar forecasts with a single model, removing the decoding stage.

Conditioning on observations. Forecasting inherently depends on available past observations. While training only for predicting L frames given C context frames is the trivial solution, more flexible conditioning is needed in practice (Schusterbauer et al., 2026b). Therefore, we first sample $k \sim \mathcal{U}(1, k_{\text{max}}), k_{\text{max}} \leq T$ and then sample a set $\mathcal{C}$ of $k = |\mathcal{C}|$ frame indices randomly without replacement. The corresponding frames are used to form clean context. Therefore, $\tau^t = 1, \forall t \in \mathcal{C}$. With probability p_U we train for unconditional generation and set $\mathcal{C} = \emptyset$. For all frames that are not in the conditioning set, we sample individual diffusion times from a distribution $p(\tau)$. Thus $\tau^t \sim p(\tau), \forall t \notin \mathcal{C}$. At inference, this permits forecasting by conditioning on past observations ($\mathcal{C} = \{1, \ldots, C\}$), infilling by excluding missing frames from $\mathcal{C}$, and conditioning on corrupted data by setting $\tau^t \neq 1$ to express confidence in individual frames.

Timestep sampling. Sampling the remaining timesteps $\tau^t, \forall t \notin \mathcal{C}$ independently would distribute them widely across different noise levels. Similar to Wewer et al. (2025) and Schusterbauer et al. (2026a) we find superior performance by concentrating them around a shared sequence-level center $\bar{\tau}$ while retaining variation across frames (Fig. 3a). We therefore sample centers uniformly $\bar{\tau} \sim \mathcal{U}(0, 1)$, and draw framewise timesteps as:

$$\tilde{\tau}^t \sim \mathcal{N}(\bar{\tau}, s_{\bar{\tau}}^2), \qquad s_{\bar{\tau}} = \min\left\{\sigma, \frac{\bar{\tau}}{2}, \frac{1 - \bar{\tau}}{2}\right\}, \qquad t \notin \mathcal{C}. \tag{3}$$

Draws outside $[0, 1]$ are replaced by uniform draws from $[0, \bar{\tau}]$ or $[\bar{\tau}, 1]$ respectively. Note that setting $\sigma = 0$ recovers shared-timestep Random Masking Video Diffusion (RaMViD) (Höppe et al., 2022)



(a) $\boldsymbol{\tau}$ sampling. (b) Timestep shift. (c) Radar corruption at $\tau = 0.4$ without timestep shifting.

Figure 3: **Training corruption.** TBar sampling couples framewise noise levels; dimensionality-aware shifting increases corruption for larger inputs.

used by FREUD (Schusterbauer et al., 2026b). Additionally, setting $p_U = 1$ and replacing Eq. (3) with an independent uniform distribution recovers Diffusion Forcing (Chen et al., 2024). Setting $\sigma > 0$ and $p_U < 1$ retains asynchronous denoising as in Diffusion Forcing and preserves clean context. We call our generalized scheme *Masked Asynchronous Diffusion* (MAD).

Dimensionality-aware corruption. As observed by Chen (2023) and Hoogeboom et al. (2023), higher-resolution inputs retain more structure at the same diffusion time τ (Fig. 3c). Stronger corruption is therefore needed at higher resolutions to maintain comparable denoising difficulty.

Prior work on pixel-space generative models (Li & He, 2025) samples diffusion timesteps from a logit-normal distribution ($\tau = \mathrm{sigmoid}(u)$, with $u \sim \mathcal{N}(\mu, \sigma^2_{\mathrm{LN}})$) and scales the noise amplitude proportionally to the image side length relative to a 256^2 px reference. We instead seek a timestep sampling scheme that explicitly accounts for both spatial and temporal resolution.

We therefore extend resolution-dependent timestep shifting (Esser et al., 2024) to the full sequence dimensionality $D = THW$ (Fig. 3b), relative to a reference base resolution $D_0 = T_0 H_0 W_0$:

$$r(\tilde{\tau}^t) = \left(\frac{D_0}{D}\right)^{(1-\tilde{\tau}^t)/2}, \qquad \tau^t = \frac{r(\tilde{\tau}^t)\tilde{\tau}^t}{1 - \tilde{\tau}^t + r(\tilde{\tau}^t)\tilde{\tau}^t}. \tag{4}$$

For $D > D_0$, this shifts intermediate towards noise, with vanishing correction towards clean data.

In summary, we first sample a sequence-level center $\bar{\tau} \sim \mathcal{U}(0, 1)$, and then draw framewise timesteps $\tilde{\tau}^t$ according to Eq. (3). We then apply dimensionality-aware shifting to obtain the final timesteps τ^t. This preserves both endpoints ($\tau = 0$ and $\tau = 1$), leaving clean context unchanged. We find that this approach outperforms rescaled logit-normal sampling for spatiotemporal forecasting.

Prediction target. As discussed by Li & He (2025), under the manifold assumption, clean data has low-dimensional structure, unlike noise and velocity targets. This makes it easier to predict with limited network capacity in high-dimensional pixel space. Since prediction and loss parameterizations can be chosen independently, following Li & He (2025), we predict the clean sequence $\hat{\mathbf{x}}_1 = f_\theta(\mathbf{x}_{\boldsymbol{\tau}}, \boldsymbol{\tau})$ and compute the velocity-space loss (Eq. (2)) through the conversion:

$$\mathbf{v}^t_\theta = \frac{\hat{\mathbf{x}}^t_1 - \mathbf{x}^t_{\boldsymbol{\tau}}}{1 - \tau^t}, \qquad t \notin \mathcal{C}, \quad \tau^t < 1. \tag{5}$$

While our method works with standard transformers, patch size imposes a trade-off between spatial detail and computational cost. Hierarchical models address this trade-off by preserving high-resolution features while moving expensive processing to lower resolutions. We therefore use an HDiT-style (Crowson et al., 2024) backbone (Fig. 4), with spatial attention in high-resolution layers and factorized spatiotemporal attention only in the lower-resolution bottleneck after an additional downsampling step.

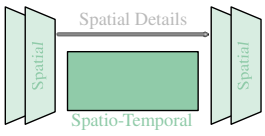


Figure 4: **HDiT backbone.**

3.3 Simple Distributional Diffusion

Flow matching models trained with Eq. (2) approximate the conditional mean velocity $\mathbf{v}^\star = \mathbb{E}[\mathbf{x}_1 - \mathbf{x}_0 \mid \mathbf{x}_{\boldsymbol{\tau}}, \boldsymbol{\tau}]$. In the limit of vanishing step size, integrating this field recovers the data distribution.

However, this no longer holds under coarse discretization with few denoising steps (De Bortoli et al., 2025). Instead, Distributional Diffusion Models (DDMs) (De Bortoli et al., 2025) learn the conditional posterior velocity $p(\mathbf{v}|\mathbf{x}_{\boldsymbol{\tau}}, \boldsymbol{\tau})$. For distributional supervision, a scoring rule over an ensemble replaces squared error regression (Fig. 5). To produce an ensemble prediction during training, the model receives auxiliary noise $\xi \sim \mathcal{N}(0, \mathbf{I})$.

For each fixed $(\mathbf{x}_{\boldsymbol{\tau}}, \boldsymbol{\tau})$, drawing $M \geq 2$ independent auxiliary noise samples produces a population of predictions (particles), $V = \{\mathbf{v}_i\}_{i=1}^{M}$, where $\mathbf{v}_i = \mathbf{v}_\theta(\mathbf{x}_{\boldsymbol{\tau}}, \boldsymbol{\tau}, \xi_i)$. In our model, these velocities are obtained from stochastic clean-data predictions through Eq. (5).

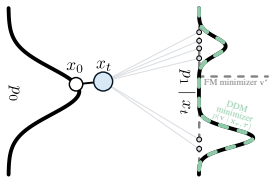


Figure 5: **DDM Training.**

From energy scores to a simple objective. For the particle population V and target velocity $\mathbf{v}$, the generalized energy score (De Bortoli et al., 2025) loss is

$$\mathcal{L}_{\lambda,\beta} = \underbrace{\frac{1}{M}\sum_i \|\mathbf{v} - \mathbf{v}_i\|_2^\beta}_{\text{confinement}} - \frac{\lambda}{2}\underbrace{\frac{1}{M(M-1)}\sum_{i\neq j}\|\mathbf{v}_i - \mathbf{v}_j\|_2^\beta}_{\text{interaction}}. \tag{6}$$

The confinement term encourages agreement with the target, while the interaction term promotes diversity. The parameters $\lambda \in [0, 1]$ and $\beta \in (0, 2]$ control interaction strength and the distance exponent, respectively. For $\lambda = 1$ and $0 < \beta < 2$, the score is strictly proper: its expectation is uniquely minimized by the target posterior. Setting $(\lambda, \beta) = (0, 2)$ recovers squared-error loss. iDDM (Anonymous, 2026) extends DDM by varying λ and β with diffusion time to favor distributional supervision at high noise levels and regression near clean data.

We aim to build a simplified distributional objective that does not require λ and β schedule tuning. Further, radar-space prediction enables the direct use of CRPS, the proper scoring rule for scalar predictive distributions commonly used in evaluation of probabilistic forecasting methods. We therefore linearly interpolate two fixed losses:

$$\boxed{\mathcal{L}_{\text{simple}}(\tau^t) = (1 - \tau^t)\,\text{CRPS} + \tau^t\,\text{MSE}.} \tag{7}$$

CRPS supervises scalar conditional marginals at each pixel, while particlewise MSE favors conditional means. The timestep-dependent interpolation transitions from marginal distributional supervision towards regression as the model approaches clean data. This is consistent with the observation that the conditional distribution becomes increasingly unimodal as $\tau \to 1$ (Biroli et al., 2024). Thus, our objective for *simple Distributional Diffusion Models* (sDDM) is an easy-to-use drop-in replacement for squared error supervision which achieves superior performance to Flow Matching and iDDM in few- and many-step settings.

Efficient training and sampling. Standard flow matching can be viewed as single-particle training: each noisy input produces one prediction. Computing the interaction term requires $M \geq 2$ predictions for the same input. Following iDDM (Anonymous, 2026), we avoid repeating the full network computation by sharing early transformer layer computations, then making M copies. To construct the particles, we concatenate independent auxiliary noise ξ_i to each copy along the token feature dimension, after modulating the noise with a learned τ-dependent gate. The remaining transformer layers process each particle separately. At inference, we use standard Euler sampling and keep context frames fixed throughout the denoising process.

4 Experiments

We use *SEVIR* (Veillette et al., 2020) and *MeteoNet* (Larvor et al., 2020) to evaluate our method. After ablating model components on the *SEVIR* (Veillette et al., 2020) benchmark, we show generalization to the *MeteoNet* (Larvor et al., 2020) data. Following standard evaluation (Gong et al., 2024; Schusterbauer et al., 2026b), we report Continuous Ranked Probability Score (CRPS), Structural Similarity Index (SSIM), Heidke Skill Score (HSS), and Critical Success Index (CSI) over 6 thresholds. Additionally, we compute rank histograms and reliability index (RI) (Delle Monache et al., 2006; Wilks, 2019) for calibration. We use 50 sampling steps and evaluate 512 samples for ablations. Metric and evaluation details are found in Appendix Sec. C. Implementation details for our method and training procedure are found in Appendix Sec. A.

Table 1: **T-sampling.** Uniform $p(\tau)$ with $T \times H \times W$ shift achieves superior performance.

T-sampling	CRPS ↓	CSI ↑	RI ↓
LN	0.0232	0.3081	0.3100
LN, noise ×1.5	0.0211	**0.3698**	0.8575
$\mathcal{U}, H \times W$	0.0210	0.3285	0.1109
$\mathcal{U}, T \times H \times W$	**0.0203**	0.3467	**0.1066**

Table 2: **Paradigm.** We compare MAD to RaMViD and Diffusion Forcing.

Paradigm	CRPS ↓	CSI ↑	RI ↓
Forcing	0.0317	0.2280	0.5301
RaMViD	**0.0203**	0.3467	0.1066
MAD uniform	0.0230	0.3380	0.2870
MAD $\sigma = 0.1$	0.0205	**0.3545**	**0.0938**

Table 3: **Architecture.** Hierarchical models with factorized attention outperform JiT.

Model B-scale	CRPS↓	CSI↑	ms / NFE ↓
JiT	0.0193	0.3782	6.55
Fact. JiT	0.0195	0.3804	**4.76**
Hierarch.	0.0186	0.3895	6.05
Fact. Hierarch.	**0.0182**	**0.4275**	5.80

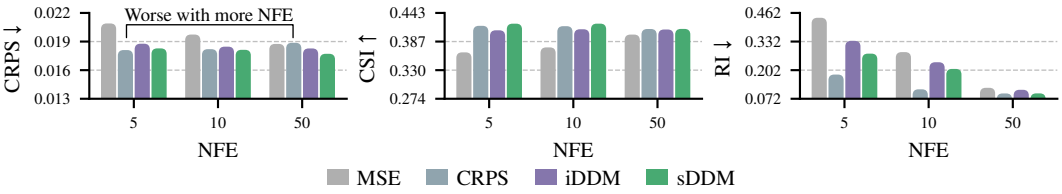


Figure 6: **Distributional Training Objective.** Training with a distributional objective unlocks few-step inference. However, using only the distributional loss diverges for high NFE.

4.1 Building JWS

The following experiments progressively build towards JWS starting from a naive pixel-space image diffusion baseline (Li & He, 2025). We use small models (24M params) unless noted otherwise. Additional ablations of hyperparameters (e.g., patch size, ensemble expansion, and training steps) are found in Appendix Sec. B.

Timestep sampling. Tab. 1 compares different $p(\tau)$ for training. We follow Schusterbauer et al. (2026b) and instantiate MAD with $\sigma = 0$ recovering RaMViD (Höppe et al., 2022). Default Logit-normal sampling (LN) fails, LN with noise scaling achieves the best CSI, yet rescaled uniform achieves superior CRPS and RI. We therefore use rescaled Uniform sampling. In Tab. 2, MAD instantiations with $p_U < 1$, which preserve clean conditioning, outperform naive Diffusion Forcing. Setting $\sigma = 0.1$, i.e. sampling correlated timesteps for predicted frames, achieves strong performance while maintaining flexibility.

Architecture. Tab. 3 shows that MAD enables effective pixel-space nowcasting with standard transformers, i.e. without specialized architectural modifications. Hierarchical models further improve performance, while factorized spatiotemporal attention reduces computational cost.

Distributional Objective. Fig. 6 compares squared error, CRPS, iDDM, and sDDM objectives. sDDM achieves strong performance for low and high NFE settings, while default squared-error Flow Matching requires many iterations and training only with CRPS deteriorates for high NFE. Fig. 7 highlights inference time scaling. Further, JWS outperforms alternative methods even with a single function evaluation. In summary, sDDM achieves strong results with limited compute and scalability to larger inference budgets.

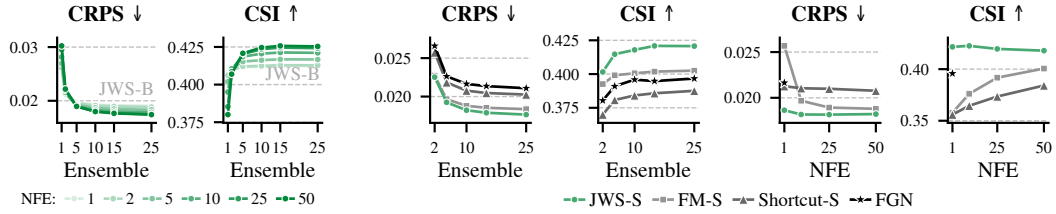


Figure 7: **Inference Time Scaling.** *Left:* JWS scales with ensemble size and NFEs. *Right:* JWS outperforms Flow Matching, Shortcut models, and FGN in matched comparisons using a forward pass.

Table 4: **SOTA Nowcasting on SEVIR.** JWS achieves superior CRPS, SSIM, HSS, and RI on SEVIR. † model trained at 128^2 for better results. ⋆ metric computed with our pipeline.

Method	Stages	Param	CRPS↓	SSIM↑	HSS↑	CSI↑	RI⋆ ↓
ConvLSTM (Shi et al., 2015) (NeurIPS '15)	1	14M	0.0264	0.7749	0.5232	0.4102	–
PredRNN (Wang et al., 2017) (TPAMI '22)	1	47M	0.0271	0.7497	0.5192	0.4045	–
PhyDNet (Guen & Thome, 2020) (CVPR '20)	1	14M	0.0253	0.7649	0.5311	0.4198	–
SimVP (Gao et al., 2022a) (CVPR '22)	1	16M	0.0259	0.7772	0.5280	0.4153	–
EarthFormer (Gao et al., 2022b) (NeurIPS '22)	1	9M	0.0251	0.7756	0.5411	0.4310	–
NowcastNet (Zhang et al., 2023) (Nature '23)	1	35M	0.0283	0.5696	0.5365	0.4152	–
PreDiff† (Gao et al., 2023) (NeurIPS '23)	2	105M	0.0202	0.7648	0.4914	0.3875	–
CasCast (Gong et al., 2024) (ICML '24)	3	402M	0.0202	0.7797	0.5602	0.4401	0.3124
FlowCast (Ribeiro & Pucer, 2026) (ICLR '26)	2	160M	0.0182	-	0.5863	**0.4651**	0.6835
FREUD (Schusterbauer et al., 2026b) (CVPR '26)	2	521M	0.0190	0.7841	0.5011	0.3864	0.1355
JWS-T/32 (*ours*)	1	9M	0.0184	0.8016	0.5132	0.3948	**0.0949**
JWS-S/32 (*ours*)	1	24M	0.0178	0.8057	0.5327	0.4116	0.0966
JWS-B/32 (*ours*)	1	94M	0.0176	0.8092	0.5408	0.4180	0.0956
JWS-L/32 (*ours*)	1	340M	**0.0175**	**0.8122**	0.5428	0.4183	0.1908
↳ with CFG = 1.5	1	340M	0.0177	0.8023	**0.5894**	0.4581	0.1815

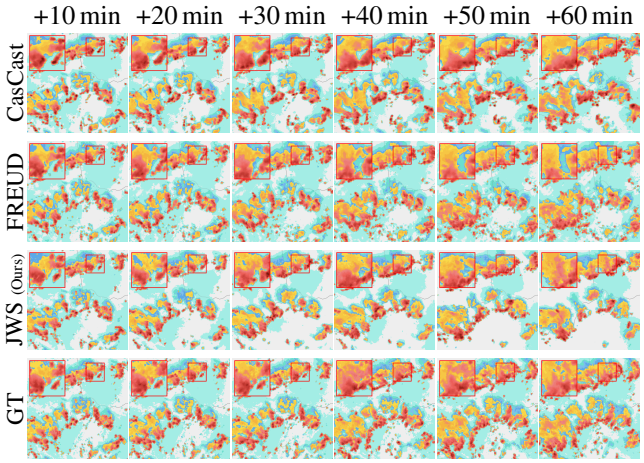


Figure 8: **Qualitative Forecast Comparison.** JWS produces sharp forecasts that align with the ground truth and remains more realistic over leadtime than FREUD and CasCast.

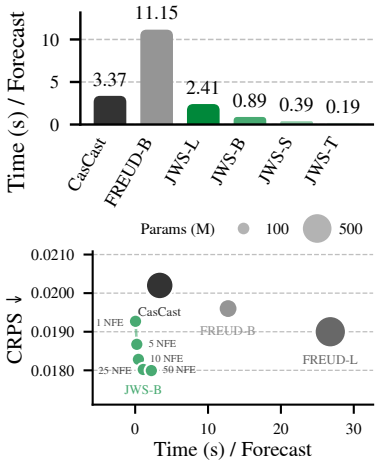


Figure 9: **Latency.** JWS enables faster inference at matched (20) NFE and strong few-step results.

4.2 System-level Weather Nowcasting

SOTA Nowcasting. In Tab. 4 JWS achieves better CRPS, SSIM, and RI on SEVIR than previous methods without classifier-free guidance (CFG). Using guidance, we additionally achieve superior HSS and perform second-best on CSI. We achieve state-of-the-art with a simplified, efficient pipeline. JWS-T performs competitively with much larger models and enables probabilistic nowcasting with a small footprint. Fig. 8 qualitatively shows sharp realistic forecasts over lead time. Additional qualitative comparisons are found in Appendix Sec. E.2. Tab. 5 evaluates JWS on MeteoNet. JWS achieves the best reported CRPS and SSIM. Guidance further improves HSS and CSI, achieving the best reported scores for these metrics. The 9M parameter JWS-T model outperforms previous methods in CRPS. Fig. 9 shows that even with 50 NFE JWS provides forecasts faster than CasCast or FREUD. One-step JWS-B produces a 10-member ensemble forecast in less than a second and performs competitively with CasCast and FREUD models.

Zero-shot sequence-length extrapolation. We evaluate zero-shot lead-time extrapolation, extending forecasts from 12 to 36 predicted frames without additional training. We exploit zero-shot deterministic conditioning following Schusterbauer et al. (2026b) to stabilize rollouts. Fig. 10 shows our method achieves superior CRPS and calibration (RI) while remaining competitive in CSI.

Table 5: **MeteoNet Nowcasting.** JWS achieves state-of-the-art nowcasting on MeteoNet. JWS-Tiny (T) outperforms $40\times$ larger models in CRPS. $\star$ metric computed with our pipeline.

Method	Stages	Param	CRPS↓	SSIM↑	HSS↑	CSI↑	RI*↓
EarthFormer (Gao et al., 2022b) (NeurIPS '22)	1	9M	0.0224	–	–	0.2831	–
NowcastNet (Zhang et al., 2023) (Nature '23)	1	35M	0.0277	–	–	0.2955	–
PreDiff (Gao et al., 2023) (NeurIPS '23)	2	105M	0.0197	–	–	0.2546	–
CasCast (Gong et al., 2024) (ICML '24)	3	402M	0.0180	–	–	0.3156	–
FREUD (Schusterbauer et al., 2026b) (CVPR '26)	2	521M	0.0193	0.7312	0.2082	0.1417	–
JWS-T/32 (*ours*)	1	9M	0.0143	0.7887	0.4026	0.2851	0.2376
JWS-S/32 (*ours*)	1	24M	0.0141	0.7917	0.4079	0.2891	0.3100
JWS-B/32 (*ours*)	1	94M	0.0144	0.7967	0.4206	0.2955	0.3858
JWS-L/32 (*ours*)	1	340M	**0.0138**	**0.7974**	0.4186	0.2959	**0.0773**
↳ with CFG = 1.5	1	340M	0.0141	0.7904	**0.4816**	**0.3448**	0.5719

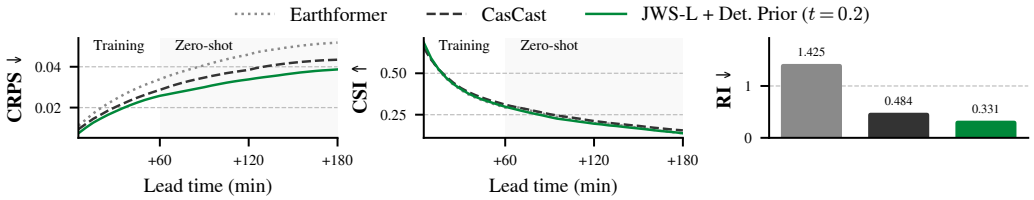


Figure 10: **Zero-shot Evaluation on SEVIR-Long.** With zero-shot deterministic conditioning, JWS shows superior zero-shot performance with 36 prediction frames up to 180 min into the future.

Calibrated Uncertainty Estimates. In Fig. 12 JWS rank histograms are flatter than for FREUD and CasCast. Calibration remains superior over lead time. In Fig. 11 ensembles at +60 min maintain accurate global structure, with local variability. Heatmaps reveal uncertainty in high precipitation regions, aligned with chaotic dynamics. Appendix Sec. E.3 shows more qualitative ensembles.

5 Conclusion

Precipitation nowcasting requires rapid updates and calibrated uncertainty. We present JWS, a simple end-to-end generative nowcasting method. JWS does not rely on compression models. Thus, ensembling quantifies uncertainty directly in radar-space. Training with a distributional objective unlocks few-step sampling at inference time, drastically reducing latency. Our approach achieves state-of-the-art probabilistic nowcasting performance on the SEVIR and MeteoNet benchmarks. Our approach is scalable with respect to the model size and the available computational budget at inference time, while efficient settings perform competitively. By greatly simplifying training and inference, we hope that JWS can be easily adopted by meteorological practitioners.

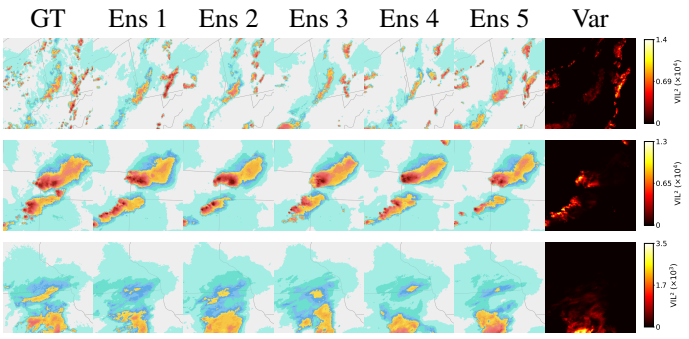


Figure 11: **Qualitative Forecast Ensemble.** At +60 min ensemble variance is aligned with precipitation intensity.

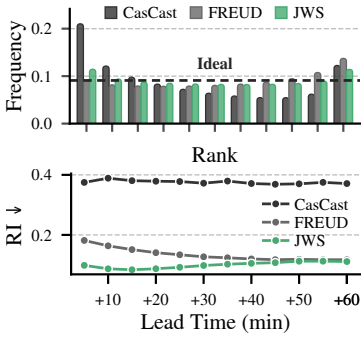


Figure 12: **Calibration.** JWS produces flatter rank histograms.

AI USE STATEMENT

In this work, we used generative AI tools for designing and providing feedback on methodology and experiments, as well as assistance with implementing methods. We have not used generative AI tools for help developing theoretical or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, assistance with translation, cleaning and reformatting datasets, support in qualitative and thematic data analysis, or interpretation of results. Synthetic data generation and assistance in writing proofs are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures, creating and editing software code, summarizing existing literature, brainstorming, sourcing/searching for information, editing the research paper for readability, and the identification of relevant literature. We have reviewed all AI-assisted work. Methodology, experiments, and brainstorming results were discussed among at least two researchers. LLM-generated code was verified and tested for correctness. LLM-assisted literature search was followed up by manual literature review and verification of the summaries by at least two researchers. Finally, AI-generated figures were manually verified and improved. Figures were discussed with at least two researchers to ensure correctness and clarity. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

JWS is trained using publicly available datasets (Veillette et al., 2020; Larvor et al., 2020). We detail training in Appendix A and data processing in Appendix C.3. JWS is a simple method that works with standard transformers (Tab. 3) requiring only minimal modifications (Sec. 3). Additionally, we detail hyperparameter settings in the Appendix in Tab. 6. We build upon existing evaluation pipelines from Gong et al. (2024) and Schusterbauer et al. (2026b) and detail evaluation protocols and metrics in Appendix Sec. C. Therefore, JWS is easy to reproduce from scratch. However, we will additionally release code, model checkpoints, and evaluation pipelines upon acceptance.

REFERENCES

Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. Machine learning for precipitation nowcasting from radar images. *arXiv preprint arXiv:1912.12132*, 2019.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Ferran Alet, Ilan Price, Andrew El-Kadi, Dominic Masters, Stratis Markou, Tom R. Andersson, Jacklynn Stott, Remi Lam, Matthew Willson, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Skillful joint probabilistic weather forecasting from marginals, 2025.

Anonymous. Improved distributional diffusion models. unpublished manuscript under review, see supplementary materials, 2026.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models, 2024.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models, 2025a.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation, 2025b.

Matthew A. Chan, Maria J. Molina, and Christopher A. Metzler. Hyper-Diffusion: Estimating Epistemic and Aleatoric Uncertainty with a Single Model, February 2024.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion, December 2024.

Shengchao Chen, Guodong Long, Jing Jiang, Dikai Liu, and Chengqi Zhang. Foundation Models for Weather and Climate Data Understanding: A Comprehensive Survey, December 2023.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow, 2025.

Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

Ting Chen and Lala Li. Fit: Far-reaching interleaved transformers, 2023.

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Xiaobin Hu, Hanzhen Zhao, Chengjie Wang, Jian Yang, and Ying Tai. Dip: Taming diffusion models in pixel space, 2026.

P. Cheung and H.Y. Yeung. Application of optical-flow technique to significant convection nowcast for terminal areas in Hong Kong. In *The 3rd WMO International Symposium on Nowcasting and Very Short-Range Forecastin*, Rio de Janeiro, Brazil, 2012. WMO.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z. Kaplan, and Enrico Shippole. Scalable High-Resolution Pixel-Space Image Synthesis with Hourglass Diffusion Transformers. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 9550–9575. PMLR, 2024.

Valentin De Bortoli, Alexandre Galashov, J Swaroop Guntupalli, Guangyao Zhou, Kevin Patrick Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=N82967FcVK.

Luca Delle Monache, Joshua P. Hacker, Yongmei Zhou, Xingxiu Deng, and Roland B. Stull. Probabilistic aspects of meteorological and ozone regional ensemble forecasts. *Journal of Geophysical Research: Atmospheres*, 111(D24):2005JD006917, December 2006. ISSN 0148-0227. doi: 10.1029/2005JD006917.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

Lasse Espeholt, Shreya Agrawal, Casper Sønderby, Manoj Kumar, Jonathan Heek, Carla Bromberg, Cenk Gazen, Rob Carver, Marcin Andrychowicz, Jason Hickey, et al. Deep learning for twelve hour precipitation forecasts. *Nature communications*, 13(1):5145, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models, 2025.

Rohit Gandikota and David Bau. Distilling diversity and control in diffusion models, 2025.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3170–3180, 2022a.

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. *Advances in Neural Information Processing Systems*, 35:25390–25403, 2022b.

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.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling, 2025.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models, 2026.

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, December 2002. ISSN 1520-0493, 0027-0644. doi: 10.1175/1520-0493(2002)130⟨2859:SDOTPO⟩2.0.CO;2.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007. ISBN: 0162-1459.

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*, volume 235 of *ICML'24*, pp. 15809–15822, Vienna, Austria, 2024. JMLR.org.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11474–11484, 2020.

Yi Guo, Wei Wang, Zhihang Yuan, Rong Cao, Kuan Chen, Zhengyang Chen, Yuanyuan Huo, Yang Zhang, Yuping Wang, Shouda Liu, and Yuxuan Wang. SplitMeanFlow: Interval Splitting Consistency in Few-Step Generative Modeling, July 2025. URL http://arxiv.org/abs/2507.16884. arXiv:2507.16884 [cs] version: 1.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood Attention Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6185–6194, 2023.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent Video Diffusion Models for High-Fidelity Long Video Generation, March 2023.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary Position Embedding for Vision Transformer, 2024.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. *arXiv preprint arXiv:2301.11093*, 2023.

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion, 2025.

Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022.

Yuhao Huang, Shih-Hsin Wang, Andrea L. Bertozzi, and Bao Wang. Rmflow: Refined mean flow by a noise-injection step for multimodal generation, 2026.

Junhwa Hur and Stefan Roth. Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation, 2019.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 14569–14589. PMLR, 2023.

Yan Ji, Bing Gong, Michael Langguth, Amirpasha Mozaffari, and Xiefei Zhi. Clgan: A gan-based video prediction model for precipitation nowcasting. *EGUsphere*, pp. 1–23, 2022.

Youssef Kossale, Mohammed Airaj, and Aziz Darouichi. Mode Collapse in Generative Adversarial Networks: An Overview. In *2022 8th International Conference on Optimization and Applications (ICOA)*, pp. 1–6, October 2022. doi: 10.1109/ICOA55659.2022.9934291.

Gwennaëlle Larvor, Léa Berthomier, Vincent Chabot, Brice Le Pape, Bruno Pradel, and Lior Perez. Meteonet, an open reference weather dataset by meteo-france. *URL https://meteonet.umr-cnrm.fr/*, 2020.

Jussi Leinonen, Ulrich Hamann, Daniele Nerini, Urs Germann, and Gabriele Franch. Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. *arXiv preprint arXiv:2304.12891*, 2023.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Hong-Bin Liu and Ickjai Lee. MPL-GAN: Toward Realistic Meteorological Predictive Learning Using Conditional GAN. *IEEE Access*, 8:93179–93186, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2995187.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, 2019.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows, 2026.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers, 2024.

Anh Nguyen, Viet Nguyen, Duc Vu, Trung Dao, Chi Tran, Toan Tran, and Anh Tran. Improved training technique for shortcut models, 2025a.

Thao Nguyen, Jiaqi Ma, Fahad Shahbaz Khan, Souhaib Ben Taieb, and Salman Khan. Raindiff: End-to-end precipitation nowcasting via token-wise attention diffusion, 2025b.

NOAA / National Centers For Environmental Information. NOAA Storm Events Database, 2020.

Dohyun Park, Changhoon Song, Tengyuan Chang, Yoo-Geun Ham, and Youngjoon Hong. Diffusion-based refinement for kilometer-scale probabilistic precipitation nowcasting, 2026.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators, February 2022.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers, 2023.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. GenCast: Diffusion-based ensemble forecasting for medium-range weather, May 2024.

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.

Bernardo Perrone Ribeiro and Jana Faganeli Pucer. Flowcast: Advancing precipitation nowcasting with conditional flow matching, 2026.

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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351, pp. 234–241. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24573-7 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28.

Hidetomo Sakaino. Spatio-Temporal Image Pattern Prediction Method Based on a Physical Model With Time-Varying Optical Flow. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5): 3023–3036, May 2013. ISSN 1558-0644. doi: 10.1109/TGRS.2012.2212201.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022.

Paola Salio, Luciana Stoll, Daniela D'Amen, Estelle De Coning, Hellen Msemo, Franziska Schmid, Solfrid Agersten, Anders Sivle, André Simon, and Maria Julia Chasco. Nowcasting and early warning systems across wmo regions associations: A pre-ew4all assessment. *Bulletin of the American Meteorological Society*, 106(12):E2490–E2508, 2025. ISSN 0003-0007, 1520-0477. doi: 10.1175/BAMS-D-24-0267.1.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023.

Johannes Schusterbauer, Ming Gui, Yusong Li, Pingchuan Ma, Felix Krause, and Björn Ommer. Denoising, fast and slow: Difficulty-aware adaptive sampling for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Johannes Schusterbauer, Jannik Wiese, Nick Stracke, Timy Phan, and Björn Ommer. Probabilistic precipitation nowcasting with rectified flow transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b.

Noam Shazeer. GLU Variants Improve Transformer, February 2020. URL `http://arxiv.org/abs/2002.05202`. arXiv:2002.05202 [cs].

Lei She, Chenghong Zhang, Xin Man, and Jie Shao. LLMDiff: Diffusion model using frozen llm transformers for precipitation nowcasting. *Sensors*, 24(18):6049, January 2024. ISSN 1424-8220. doi: 10.3390/s24186049.

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.

Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in neural information processing systems*, 30, 2017.

Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. Metnet: A neural weather model for precipitation forecasting. *arXiv preprint arXiv:2003.12140*, 2020.

Changhoon Song, Teng Yuan Chang, and Youngjoon Hong. Extreme weather nowcasting via local precipitation pattern prediction, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023.

Simone Sperati, Stefano Alessandrini, and Luca Delle Monache. An application of the ECMWF Ensemble Prediction System for short-term solar power forecasting. *Solar Energy*, 133:437–450, August 2016. ISSN 0038-092X. doi: 10.1016/j.solener.2016.04.016. URL https://www.sciencedirect.com/science/article/pii/S0038092X1630041X.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, 2023.

Mark Veillette, Siddharth Samsi, and Chris Mattioli. SEVIR : A Storm Event Imagery Dataset for Deep Learning Applications in Radar and Satellite Meteorology. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22009–22019. Curran Associates, Inc., 2020.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion, 2025.

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.

Christopher Wewer, Bart Pogodzinski, Bernt Schiele, and Jan Eric Lenssen. Spatial reasoning with denoising models, 2025.

D. S. Wilks. Indices of Rank Histogram Flatness and Their Sampling Properties. *Monthly Weather Review*, 147(2):763–769, February 2019. ISSN 0027-0644, 1520-0493. doi: 10.1175/MWR-D-18-0369.1.

Wang-chun Woo and Wai-kin Wong. Operational Application of Optical Flow Techniques to Radar-Based Rainfall Nowcasting. *Atmosphere*, 8(3):48, February 2017. ISSN 2073-4433. doi: 10.3390/atmos8030048.

Yuan-Kang Wu, Po-En Su, Ting-Yi Wu, Jing-Shan Hong, and Mohammad Yusri Hassan. Probabilistic Wind-Power Forecasting Using Weather Ensemble Models. *IEEE Transactions on Industry Applications*, 54(6):5609–5620, November 2018. ISSN 0093-9994, 1939-9367. doi: 10.1109/TIA.2018.2858183.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization, 2019.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation, 2024b.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27758–27767, 2024.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation, 2025.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/1e8a19426224ca89e83cef47f1e7f53b-Abstract.html.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 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.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching, 2025.

Yufeng Zhu, Chunlei Shi, Yongchao Feng, and Dan Niu. Pixelflowcast: Latent-free precipitation nowcasting via pixel mean flows, 2026.

SUPPLEMENTARY MATERIALS

A IMPLEMENTATION DETAILS

We summarize architecture and training hyperparameters in Tab. 6.

Table 6: **Hyperparameters.** Settings of training runs for JWS across model size.

Parameter	JWS-T	JWS-S	JWS-B	JWS-L
Parameters (M)	9	24	94	340
High-resolution layers (down & up)	2	2	2	2
Bottleneck	6	8	8	20
High resolution hidden dimension	192	256	512	768
Bottleneck hidden dimension	256	384	768	1024
Attention heads (HiRes/Bottleneck)	3/4	4/6	8/12	12/16
Normalization	RMSNorm (Zhang & Sennrich, 2019)			
FFN expansion factor	$3\times$			
Activations	SwiGLU (Shazeer, 2020)			
Positional embeddings	3D Axial RoPE (Su et al., 2023; Heo et al., 2024)			
Diffusion time mapping dimension	128	128	384	384
Diffusion time mapping depth	2	2	2	2
Diffusion time conditioning	AdaNorm (Peebles & Xie, 2023; Xu et al., 2019)			
Input patch size $(T \times H \times W)$	$1 \times 16 \times 16$			
Total bottleneck patch size $(T \times H \times W)$	$1 \times 32 \times 32$			
High resolution attention window $(T \times H \times W)$	$1 \times 5 \times 5$			
Tbar σ	0.1			
Tbar sampling	Uniform + $T \times H \times W$ shift			
Maximum conditioning frames $k_{\max}$	13			
Unconditional probability p_U	0.25			
Training particles	3	3	3	3
Expansion before layer	8	10	10	22
ξ embedding dimension	256			
ξ conditioning	Timestep-gated residual concatenation (Anonymous, 2026)			
Training frame resolution	384×384			
Training sequence length	25			
SEVIR training sequences	402,500 (16,100 events $\times$ 25 windows)			
Optimizer	AdamW (Loshchilov & Hutter, 2019)			
AdamW (β_1, β_2)	$(0.9, 0.999)$			
Learning rate	10^{-4}			
Learning-rate schedule	Constant with 1,000 warmup steps			
EMA decay	0.9999			
Precision	BF16 mixed precision			
Global Batch size	128			
Training GPUs	8	8	8	16
Training steps	300k	300k	200k	200k
Training schedule	N/A	N/A	N/A	100k on 128^2 +finetuning

Architecture. We build JWS using standard Llama-style Transformers. Model variants across scales differ only in the number of layers and their hidden dimensionality, resulting in four model sizes ranging from 9M to 340M parameters. All hierarchical models use an initial patchification of $1 \times 16 \times 16$ matching Li & He (2025). We follow Schusterbauer et al. (2026b) and use neighborhood attention (Hassani et al., 2023) for the high-resolution transformer layers, restricting attention to a 5×5 region within each frame. All models use two high resolution layers before down- and two layers after upsampling (four in total). The remaining layers are applied after additional $1 \times 2 \times 2$ patchification ($1 \times 32 \times 32$ in total). Bottleneck layers attend to the full spatiotemporal grid but use

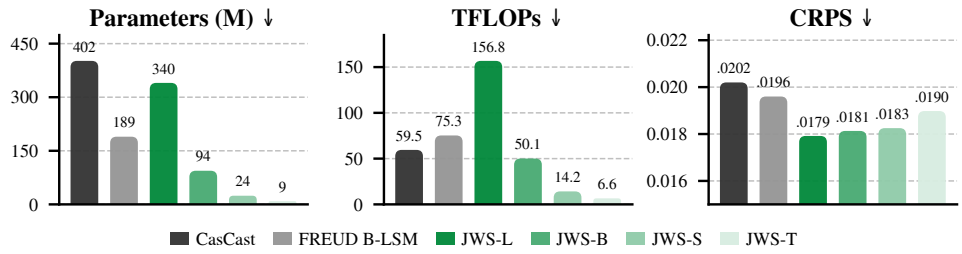


Figure 13: **FLOPs vs. Performance.** Our JWS-B, JWS-S, andJWS-T models are significantly more compute and parameter efficient while achieving superior nowcasting performance.

interleaved factorized attention, interleaving only-spatial and only-temporal attention blocks. All models use Adaptive RMSNorm (Zhang & Sennrich, 2019; Peebles & Xie, 2023; Xu et al., 2019) for normalization and diffusion-time conditioning. Further, all models use SwiGLU (Shazeer, 2020) activations and 3D Axial Rotary Positional Embeddings (Su et al., 2023; Heo et al., 2024).

All variants expand the ensemble for distributional training before the last bottleneck layer. All following layers (bottleneck and high-resolution) receive noise conditioning. The auxiliary noise ξ is sampled with dimensionality 256 and concatenated to the residual stream after modulation by a diffusion time-dependent gate following (Anonymous, 2026).

Training. During training, we sample $\bar{\tau}$ uniformly and draw framewise timesteps using Eq. (3) with standard deviation cap $\sigma = 0.1$. We then apply dimensionality-aware shifting using the full spatiotemporal resolution $T \times H \times W$. We sample a maximum of 13 clean conditioning frames during training and use fully unconditional training with $\mathcal{C} = \emptyset$ with probability $p_U = 0.25$, enabling classifier-free guidance (Ho & Salimans, 2022) at inference time.

We train with $25 \times 384 \times 384$ sequences from the SEVIR (Veillette et al., 2020) dataset ($24 \times 400 \times 400$ for MeteoNet). Exhaustive subsequence sampling following Schusterbauer et al. (2026b) yields 402,500 total training sequences (259,656 for MeteoNet). JWS-T and JWS-S are trained for 300k iterations using the AdamW (Loshchilov & Hutter, 2019) optimizer (lr $= 10^{-4}$, betas $= (0.9, 0.999)$) and a global batch size of 128. Due to overfitting on small data we train JWS-B only for 200k iteration and train JWS-L only for 100k iterations after initializing from a checkpoint pretrained on smaller crops. We provide further information on overfitting and show scalability with more data in Sec. B. We use Exponential Moving Average (EMA) weights to stabilize training. For MeteoNet training we resize frames to 384×384 resolution to use the matching model configurations across datasets, while restoring 400×400 for metric calculation via bilinear upsampling.

Inference. During inference, we integrate the ODE using standard Euler updates. We keep the auxiliary noise ξ fixed for the denoising trajectory. Further, we do not use particle expansion and instead mirror prior work (Leinonen et al., 2023; Gao et al., 2023; Ribeiro & Pucer, 2026; Schusterbauer et al., 2026b) and sample different noises $\mathbf{x}_0$ to ensemble forecasts. We use 50 NFE by default.

B Additional Ablations

Performance over Lead time. In addition to Fig. 10, we also evaluate performance over lead time for the in-domain setting. Fig. 24 shows our model achieves superior CRPS and SSIM throughout the entire forecast horizon with advantages increasing for longer forecasts. HSS remains superior to FREUD (Schusterbauer et al., 2026b) and is competitive with CasCast (Gong et al., 2024) which relies on deterministic priors and classifier-free guidance to improve localization-centric metrics.

FLOP comparison. In addition to wall clock results in Fig. 9, Fig. 13 shows model capacity and inference FLOPs in relation to resulting CRPS. For the configurations shown, our smallest model requires 6.6 TFLOPs compared with CasCast's 59.5 TFLOPs, an approximately ninefold reduction. Our models achieve superior CRPS with substantially lower parameter count and computational

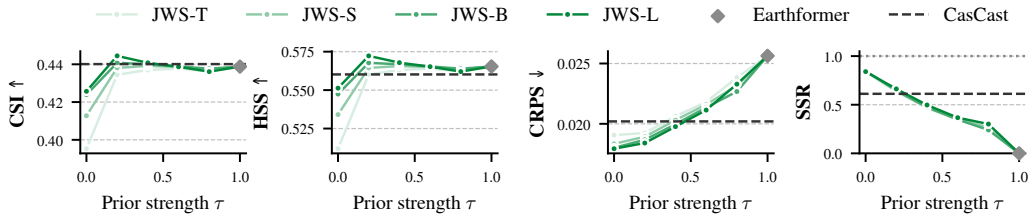


Figure 14: **Deterministic Conditioning.** Conditioning on a deterministic prior improves localization-centric CSI and HSS but negatively impacts calibration consistently.

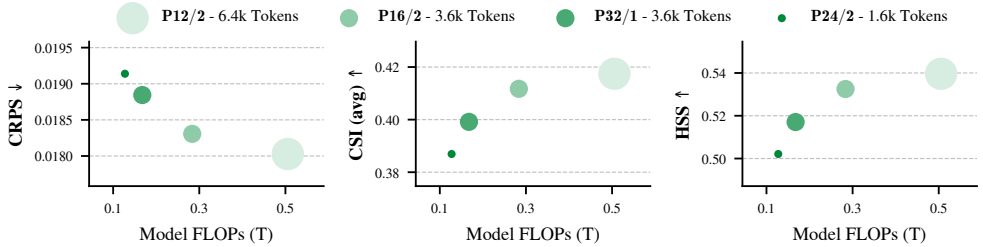


Figure 15: **Patch Size Ablation.** Varying patch sizes at both hierarchy levels changes the token count and thus increases FLOPs. More tokens/FLOPs improve CRPS, CSI, and HSS.

cost. Note that JWS-L requires more inference FLOPs than CasCast but is faster in wall clock time, highlighting the benefit of reduced pipeline overhead.

Zero-shot deterministic conditioning. Following Schusterbauer et al. (2026b), we explore zero-shot deterministic conditioning by initializing denoising from noised Earthformer (Gao et al., 2022b) predictions. The noising timestep t controls the strength of the prior. In Fig. 14 weak priors improve CSI and HSS up to $t = 0.2$. However, deterministic conditioning consistently worsens calibration. MAD enables using different diffusion timesteps per frame to express time-dependent confidence in a prior. Using small JiT models, Fig. 19 shows that using higher diffusion time (cleaner priors) for early frames and lower diffusion time (noisier priors) for later frames outperforms uniform time for CRPS, reliability and CSI. The RaMViD baseline underperforms for calibration.

Patch size. Reducing the patch size at either hierarchy level increases the number of processed tokens. Fig. 15 shows a clear scaling trend. Processing more tokens results in superior forecasts.

Scalability. Both Tab. 4 and Tab. 5 establish the scalability of JWS. Fig. 16 further shows that larger models consistently reach better performance earlier in training. Due to limited data availability, we initialize the L-model from a checkpoint pretrained on random 128^2 crops to mitigate overfitting (Fig. 17) and use fewer training steps for JWS-B and JWS-L. However, artificially in-

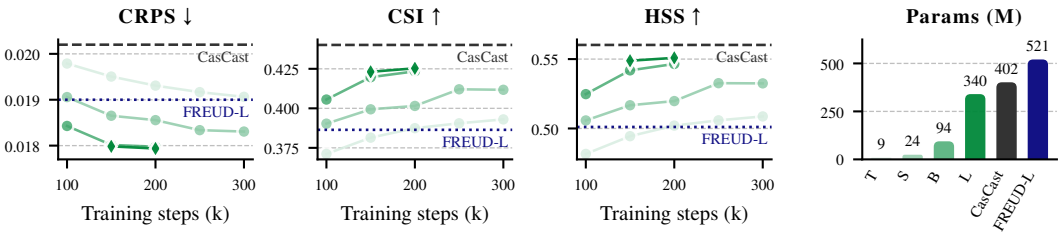


Figure 16: **Model Scalability.** Our smallest *Tiny* model already outperforms FREUD LSM-L (Schusterbauer et al., 2026b). With increased model scale, our model shows superior results.

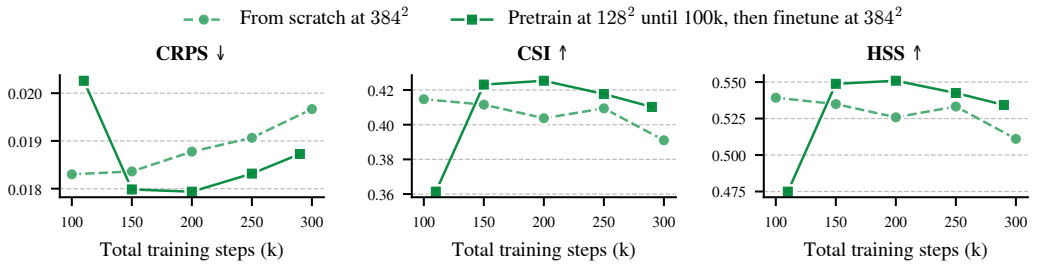
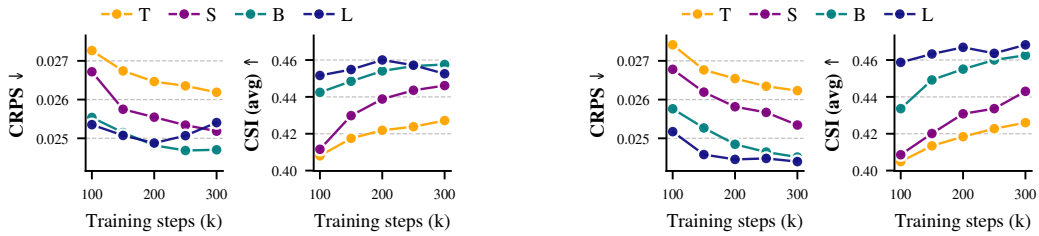


Figure 17: **Overfitting.** Training large models directly on full resolution data leads to strong overfitting. Pretraining on downsampled data first preserves scalability.



(a) **50% Data** (b) **100% Data**

Figure 18: **Data Scaling.** Training on random crop-augmented data JWS-L achieves stable training. Reducing the dataset size induces overfitting.

creasing the dataset size by random cropping enables scalable training. In Fig. 18 we train all model variants with random crops to 192×192 resolution and use the full set of events or only 50% of the available data. On reduced data, the large model still shows overfitting, while training remains stable on the full dataset. This establishes the scalability of JWS if sufficient training data is available.

Sampler. Fig. 20 compares different integration methods across inference budgets. We evaluate standard Euler sampling, the distributional sampler by De Bortoli et al. (2025), Heun sampling, and SDE integration following the protocol of Ma et al. (2024). We find strong performance with standard Euler, outperforming Heun and SDE across all metrics and budgets. The distributional sampler achieves superior SSIM and HSS, indicating superior sharpness, but underperforms Euler in CRPS evaluation. We use Euler for simplicity.

Number of Training Particles. Using larger training ensembles (more particles) increases computational cost. Yet, Fig. 21 does not reveal a scaling trend. While 6 training particles achieve superior few-step performance, the model performs worse for high NFE. The four-particle model underperforms compared to using three particles.

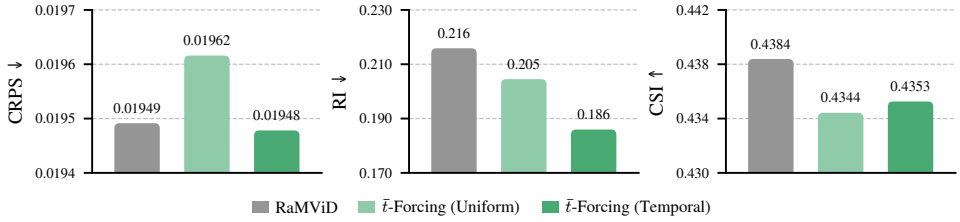


Figure 19: **Confidence-aware priors.** Adjusting prior t based on timesteps improves performance.

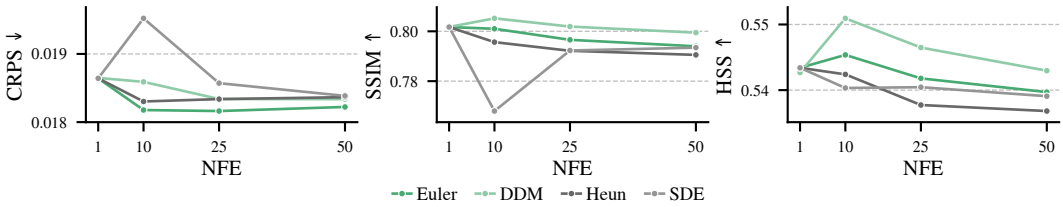


Figure 20: **Different Solvers.** Standard Euler sampling outperforms Heun and SDE integration. While DDM sampling achieves superior HSS and SSIM, Euler performs best for CRPS.

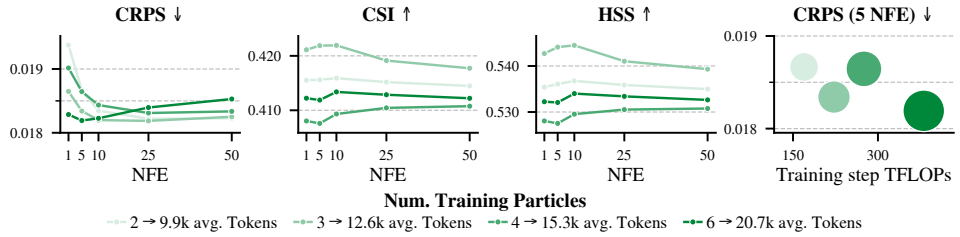


Figure 21: **Training Particles.** Increasing training particles does not improve performance.

Expansion layer. Earlier expansion of the ensemble increases computational cost during training. Matching Anonymous (2026), Fig. 22 does not show improved performance by expanding earlier. Instead, ensemble expansion at the last bottleneck layer seems to provide superior performance.

Additionally, we conducted an ablation on downsampled $128\,\mathrm{px}^2$ SEVIR data with an S-scale hierarchical model. Fig. 23 shows heatmaps for the full expansion layer $\times$ training particles grid. Again, there is no consistent impact of more particles or earlier expansion on final performance.

Auxiliary conditioning. On the $128\,\mathrm{px}^2$ SEVIR data, Tab. 7 evaluates different mechanisms for ξ injection. We compare concatenation, addition, repeated addition in every layer, and residual concatenation (Anonymous, 2026). Matching iDDM, concatenation to the residual stream is superior. Additionally, we test keeping final layers clean, which deteriorates performance.

$\bar{\tau}$ spread. Tab. 8 compares different spread values σ for MAD sampling. $\sigma = 0.1$ outperforms other spread values, including uniform timesteps and uncorrelated timesteps.

Loss Reweighting and Data Filtering. Previous work (Gong et al., 2024) filters MeteoNet and trains only on sequences with observed precipitation. In addition to "all data" we test (with small JiT models): filtering, binary loss weighting, downweighting dry pixels based on their frequency in the sample, and power-law loss weighting. However, Tab. 9 and Tab. 10 show that neither filtering nor reweighting improves overall performance, while all such interventions deteriorate CRPS. We therefore require no dataset filtering or loss reweighting and train directly on the full training dataset with equally weighted samples and pixels.

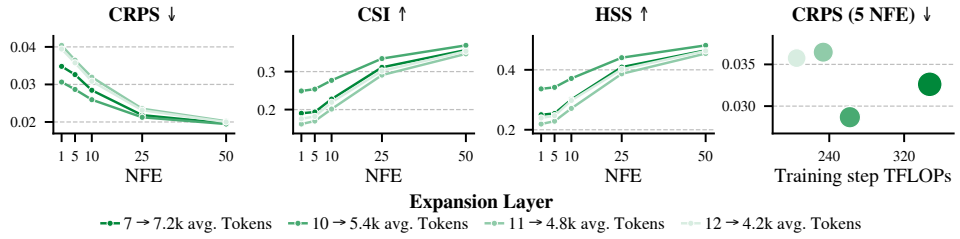
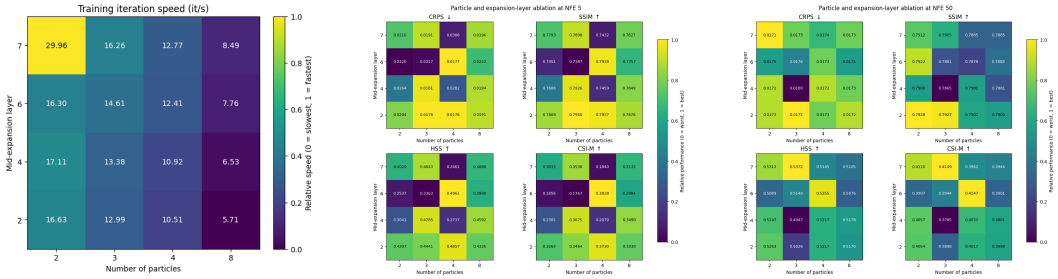


Figure 22: **Expansion Layer.** Expanding the ensemble earlier does not yield improved performance.



(a) Throughput. (b) 5 NFE. (c) 50 NFE.

Figure 23: **Small-scale verification.** We again find no gain using more particles or earlier expansion.

Table 7: **Conditioning Path.** Concatenating noise to the residual stream is best.

ξ conditioning	CRPS ↓	HSS ↑	CSI-M ↑
Single concatenation	0.0236	0.4021	0.3023
Residual concatenation	**0.0217**	**0.4351**	**0.3295**
Residual concatenation + 2 clean	0.0276	0.3895	0.2916
Single addition	0.0238	0.4176	0.3125
Repeated addition	0.0243	0.4010	0.3028
Repeated addition + 2 clean	0.0245	0.4118	0.3081

Table 8: **Tbar σ ablation.** Setting $\sigma = 0.1$ achieves the best result.

σ	CRPS ↓	HSS ↑	CSI-M ↑
0 (RaMViD (Höppe et al., 2022))	0.01940	0.4740	0.3638
0.05	0.01921	0.5072	0.3889
0.1	**0.01895**	**0.5154**	**0.3965**
0.2	0.01940	0.4886	0.3750
✗ (RaMViD Forcing)	0.02771	0.3875	0.2826

Table 9: **Reweighting SEVIR.** Data filtering and loss weighting have minimal impact.

Method	CRPS ↓	SSIM ↑	HSS ↑	CSI-M ↑	RI ↓
Baseline	**0.02003**	0.7653	0.5034	0.3869	0.0993
Filtering	0.02086	0.7416	**0.5087**	**0.3908**	**0.0712**
Power law	0.02016	**0.7680**	0.4691	0.3587	0.1077
Binary	0.02015	0.7669	0.4773	0.3652	0.0928

Table 10: **Reweighting MeteoNet.** Reweighting and resampling increase CRPS.

Method	CRPS ↓	SSIM ↑	HSS ↑	CSI-M ↑	RI ↓
Baseline	**0.01552**	**0.7407**	0.3942	0.2787	0.3640
Filtering	0.01586	0.7307	**0.4107**	**0.2920**	0.4302
Power law	0.01557	0.7353	0.4062	0.2885	0.3918
Binary	0.01554	0.7386	0.3997	0.2838	**0.3625**

C Metrics and Evaluation Protocol

To enable better interpretability of our results, we summarize metrics and the evaluation protocol below.

C.1 Metric Definitions

We use the metrics implemented by Schusterbauer et al. (2026b) which adopts the evaluation by Gong et al. (2024).

CRPS. CRPS is a proper scoring rule for univariate regression that generalizes the Mean Absolute Error to probabilistic tasks (Gneiting & Raftery, 2007). For observed value x, cumulative forecast distribution $F(y)$, and ensemble members $\{f_1, ..., f_N\}$ CRPS is defined as

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

The first term pushes the expectation to the ground truth, while the second term encourages spread.

SSIM. The Structural Similarity Index Measure is used to measure structural similarity between ground truth and mean forecasts. For single-channel data SSIM is defined as

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

where x is the ground truth and y is the prediction, μ_x and μ_y are the average intensities, σ_x^2 and σ_y^2 are the variances of intensities, σ_{xy} is the covariance between the images and C_1 and C_2 are two small constants. Following standard practice, SSIM is calculated by averaging sliding window results and using Gaussian weighting to calculate mean and variance values.

HSS. Heidke Skill Score (HSS) is computed on a per-pixel basis to identify positional accuracy. Using a user-defined threshold, true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) are first calculated based on binarized ground truth and forecasted precipitation. Then HSS is computed as

$$HSS = \frac{2(TP \cdot TN - FP \cdot FN)}{(TP + FN)(FN + TN) + (TP + FP)(FP + TN)}. \tag{8}$$

Following prior work we average HSS over six thresholds (16, 74, 133, 160, 181, 219) to obtain the final value. Intuitively, HSS indicates improvement over random chance, where 0 indicates no improvement and 1 corresponds to perfect prediction.

CSI. The Critical Success Index (CSI) or Threat Score quantifies the proportion of correctly predicted events excluding true negatives. Therefore, it remains useful when non-events dominate the data as in precipitation forecasting. After the same binarization as for HSS, CSI is given by

$$CSI = \frac{TP}{TP + FP + FN}. \tag{9}$$

The reported CSI is averaged over the same thresholds as HSS.

RI. To compute the reliability index (RI) we first compute rank histograms by inserting observed pixel values into a sorted list of ensemble predictions and counting the rank. Therefore, for an N-member ensemble, the rank histogram shows $N + 1$ possible ranks, resolving ties by random sampling. RI then measures the deviation of the resulting relative frequencies (p_r) from a uniform distribution as

$$\text{RI} = \sum_{r=0}^{N} \left| p_r - \frac{1}{N+1} \right|. \tag{10}$$

Thus, lower values indicate closer agreement with uniform rank distribution, i.e. superior calibration.

SSR. Spread-Skill Ratio (SSR) compares the ensemble spread with root mean squared error. SSR is computed as

$$\text{SSR} = \frac{\sqrt{\sigma^2}}{\sqrt{(\bar{f} - x)^2}}, \tag{11}$$

where $\bar{f} = N^{-1} \sum_{i=1}^{N} f_i$ is the ensemble mean, $\sigma^2 = (N-1)^{-1} \sum_{i=1}^{N} (f_i - \bar{f})^2$ is the sample variance. A calibrated ensemble forecast should have high variance when uncertain and therefore most wrong. Thus, calibration is indicated by SSR close to 1.

C.2 Evaluation Protocol

Unless indicated otherwise we use 50 network evaluations for our method and compare all approaches using 10 ensemble members. Experiments are mainly conducted on the SEVIR dataset (Veillette et al., 2020) using the train-test split provided by Gong et al. (2024). We use the full test set to establish state-of-the-art in Tab. 4 and Tab. 5. However, following prior work (Schusterbauer et al., 2026b; Ribeiro & Pucer, 2026) we use a subset of 512 samples for plots and ablation experiments due to computational constraints. Tab. 11 shows that 512 sample evaluation closely matches results with more samples and exhibits low inter-seed variance. All evaluations are carried

Table 11: **Metric Sensitivity.** Results report mean ± standard deviation across eight evaluation seeds.

Num. Samples	CRPS ↓	SSIM ↑	HSS ↑	CSI ↑	RI ↓
64	0.0259 ± 0.0025	0.6374 ± 0.0143	0.3379 ± 0.0115	0.2529 ± 0.0093	0.0615 ± 0.0183
128	0.0257 ± 0.0013	0.6372 ± 0.0123	0.3384 ± 0.0102	0.2527 ± 0.0078	0.0562 ± 0.0106
256	0.0257 ± 0.0008	0.6363 ± 0.0065	0.3365 ± 0.0077	0.2515 ± 0.0054	0.0558 ± 0.0064
512	0.0260 ± 0.0005	0.6331 ± 0.0050	0.3371 ± 0.0055	0.2521 ± 0.0039	0.0573 ± 0.0039
1024	0.0260 ± 0.0004	0.6322 ± 0.0028	0.3363 ± 0.0033	0.2511 ± 0.0027	0.0581 ± 0.0020

out on 4 Nvidia GH200 GPUs using parallelization across samples for speedup, while latency is computed with a single GPU.

For baselines we use the results reported in the literature (Gong et al., 2024; Schusterbauer et al., 2026b; Ribeiro & Pucer, 2026) after verifying reproducibility with our pipeline where possible. Metrics that were previously not reported are recomputed using available checkpoints if trained weights are provided by the authors.

C.3 Datasets

SEVIR. Our main evaluations are carried out using the SEVIR (Veillette et al., 2020) dataset. SEVIR contains 20,393 weather events of which 20% correspond to known extreme weather events from the NOAA Storm Event Database (NOAA / National Centers For Environmental Information, 2020). The remaining 80% were randomly chosen while oversampling high precipitation. SEVIR is collected from 2017–2019 over the continental United States region. Each sample covers a $384\,\text{km} \times 384\,\text{km}$ area for 4 h with 1 km spatial and 5 min temporal resolution. Precipitation is indicated by the NEXRAD Vertically Integrated Liquid product (VIL).

We use the train-test split provided by Gong et al. (2024) and follow Schusterbauer et al. (2026b) in joining the validation set into training and sampling exhaustive subsequences to increase training data size. Following prior work, we use 65 min (13 frames) conditioning to predict 60 min (12 frames) future precipitation unless noted otherwise. We follow the standard preprocessing (Veillette et al., 2020) and keep non-linear VIL encoding to $[0, 255]$ which we normalize further to $[-1, 1]$ for training following Schusterbauer et al. (2026b). To ensure comparability (Gong et al., 2024; Gao et al., 2022b; She et al., 2024; Schusterbauer et al., 2026b; Ribeiro & Pucer, 2026), we use $H = W = 384$, $C = 13$, $L = 12$, and $T = C + L = 25$ unless specified otherwise.

MeteoNet. The MeteoNet benchmark (Larvor et al., 2020) includes observations over fixed regions in northwestern and southeastern France from 2016–2018. Radar measurements are provided in 5 min intervals with $\approx 1\,\text{km}^2$ resolution. Following Gong et al. (2024), we use the old radar reflectivity product and extract the top-left 400×400 grid cells of the southeastern region. Again, we combine training and validation periods and exhaustively sample subsequences for training, while testing on the same data as prior work (Gong et al., 2024; Schusterbauer et al., 2026b). Following Gong et al. (2024) we use 24-frame subsequences and predict the next 12 frames given a 12-frame context. We normalize the reflectivity values from $[0, 70]$ to $[-1, 1]$ for training. To enable using the same JWS architecture as for SEVIR, we bilinearly resize to 384×384 for model processing, but keep metric computation at the native resolution. To compute HSS and CSI we use thresholds of $[19, 25, 35, 40, 47]$ dBZ (Gong et al., 2024).

D Extended Related Work

We expand on the related works presented in Sec. 2 to provide additional context on existing methods and our advances upon these.

D.1 Machine Learning for Nowcasting

Weather nowcasting refers to short-term, high-resolution forecasting over lead times ranging from one to twelve hours. Typically nowcasting operates on ≤ 5 km spatial grids with ≤ 10 min temporal resolution. Physics-based numerical weather prediction is computationally expensive, and therefore not well-suited for applications that require rapid updates (Shi et al., 2015; 2017; Sønderby et al., 2020; Leinonen et al., 2023; Ravuri et al., 2021; Chen et al., 2023). Early research used optical flow based forecasts (Woo & Wong, 2017; Cheung & Yeung, 2012; Germann & Zawadzki, 2002; Sakaino, 2013) for rapid-update tasks, however these approaches cannot model rain onset or dissipation (Hur & Roth, 2019; Shi et al., 2015). Therefore, machine learning methods based on convolutional neural networks (Shi et al., 2015; Agrawal et al., 2019), recurrent networks (Wang et al., 2017), or transformers (Gao et al., 2022a;b) have been developed. These approaches use patterns in observed past context to regress future weather states. While machine learning methods also begin to dominate medium- and long-range forecasting (Sønderby et al., 2020; Espeholt et al., 2022; Pathak et al., 2022; Price et al., 2024; Alet et al., 2025), they are especially important in nowcasting due to their significantly reduced inference time.

However, weather dynamics are inherently chaotic and unpredictable (Ravuri et al., 2021) requiring probabilistic forecasting under partially observable initial conditions. Yet, deterministic regression models lack robust uncertainty quantification and average possibilities leading to increasingly blurry forecasts over lead time. Alet et al. (2025) proposed Functional Generative Networks (FGNs) for medium-range forecasting, training a deterministic model to predict an ensemble of forecasts directly by optimizing an ensemble scoring function. However, they restrict the variability of forecasts to be encoded in a low-dimensional noise vector. Local variability in nowcasting is typically higher than variance in medium-range forecasting limiting the applicability of this approach compared to diffusion models due to reduced distributional coverage. Therefore, FGNs still become increasingly blurry in our experiments (Appendix E.2).

Therefore, researchers employed generative adversarial (Ravuri et al., 2021; Zhang et al., 2023; Liu & Lee, 2020) and generative diffusion (Leinonen et al., 2023; Gao et al., 2023; Ribeiro & Pucer, 2026) methods, trained to produce a sample from the conditional distribution of predictions rather than to regress the mean. As adversarial approaches are prone to mode collapse (Kossale et al., 2022) and, thus, fail to capture the full posterior distribution, diffusion-based nowcasting has become the de facto standard approach. Early diffusion-based precipitation nowcasting approaches sampled future evolution based only on prior observations (Leinonen et al., 2023; Gao et al., 2023). However, later papers argued conditioning the model on a deterministic forecast (Gong et al., 2024) or sampling only the residual between a deterministic forecast and the ground truth (Yu et al., 2024) enables superior forecasting. Similarly, Park et al. (2026) build upon the deterministic exPreCast model (Song et al., 2026) by using a diffusion model to model only small scale variability and enable ensembling of stochastic elements. However, FREUD (Schusterbauer et al., 2026b) argues conditioning on a deterministic forecast biases the generation and thus limits uncertainty quantification. Compounding evidence by Ribeiro & Pucer (2026) supports the claim that training a sufficiently capable diffusion model suffices for accurate forecasting and superior calibration.

An additional downside is that most diffusion-based nowcasting methods (Leinonen et al., 2023; Gao et al., 2023; Gong et al., 2024; Ribeiro & Pucer, 2026; Schusterbauer et al., 2026b) operate in a compressed latent space to enable faster sampling. However, compression is inherently lossy and thus induces additional uncertainty about reconstructions. Schusterbauer et al. (2026b) propose to quantify this uncertainty using generative reconstructions. However, this means for each latent

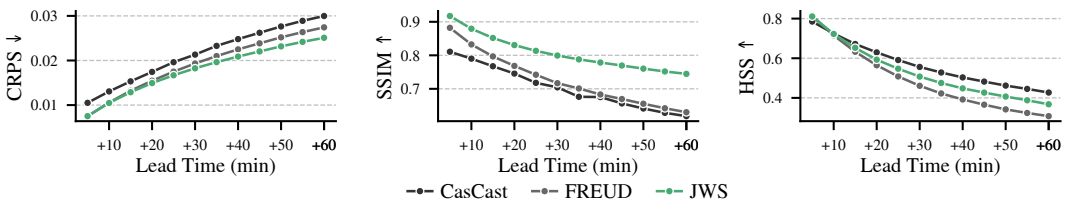


Figure 24: **Performance over lead time.** JWS-B outperforms CasCast and FREUD across the entire lead time range for CRPS and SSIM, while achieving competitive performance for HSS.

ensemble member a full ensemble of reconstructions has to be constructed to quantify all uncertainty, inducing a significant computational overhead. Some previous works instead tackle pixel-space nowcasting using specialized architectures (Nguyen et al., 2025b) or by only modeling residuals from a deterministic forecast (Yu et al., 2024; Zhu et al., 2026) which still requires a two-stage design and biases forecasts. In comparison, JWS enables pixel space nowcasting with a standard, scalable architecture and with a single end-to-end model, simplifying training and inference.

D.2 Pixel-space Diffusion

Initial works in diffusion-based image generation applied iterative denoising in pixel-space (Dhariwal & Nichol, 2021; Ho et al., 2020). Yet, computational costs made these models difficult to scale and deploy. Latent diffusion (Rombach et al., 2022) proposed to first compress perceptual information into a low-dimensional latent and then apply the diffusion process only in this compact latent space. This enabled high-resolution image generation on consumer hardware and training video generation models with academic resources (He et al., 2023).

However, the complexity of training multiple stages motivated the authors of simple diffusion (Hoogeboom et al., 2023) to develop an architecture for direct pixel-space diffusion modeling by adapting the noise schedule and downsampling early in the UNet (Ronneberger et al., 2015) backbone to concentrate capacity in the bottleneck. Further improvements (Hoogeboom et al., 2025) identify reducing skip connections as a pathway to further enhance scalability. Through this modification, improved loss weighting, and preferring scaling the number of processed patches over increasing the model capacity they achieve strong pixel-space generation performance.

Alternatives like PixelFlow (Chen et al., 2025) propose to modify the denoising process itself to enable pixel-space modeling. Chen et al. (2025) propose to schedule the image resolution over the denoising trajectory, progressively moving from noisy low resolution images to clean high resolution images during sampling. Thus, their approach saves compute on high-frequency details while these are masked by noise. Wang et al. (2025), on the other hand, enable high-resolution image synthesis by predicting patch-specific neural-field weights, with the resulting fields mapping noisy pixel values and spatial coordinates to pixel-wise velocities. This concentrates expensive transformer computation on low resolution patches while using lightweight neural fields for pixel-level prediction, avoiding the cost of processing each pixel as a separate transformer token.

Alternatively, both PixelDiT (Yu et al., 2025) and DiP (Chen et al., 2026) propose combining a large transformer for global structure with local refinement modules to enable pixel-space high resolution image generation. Other alternative architectures such as RIN (Jabri et al., 2023) and FIT (Chen & Li, 2023) enable high-resolution pixel-space modeling by reducing computational complexity of expensive self-attention. RIN (Jabri et al., 2023) concentrates computation on a compact set of latent tokens that exchange information with pixel-space data tokens through cross-attention, reusing latent representations across denoising steps. Chen & Li (2023) propose interleaving local attention within groups of data tokens with global attention over a smaller subset of tokens.

However, Li & He (2025) show that standard Vision Transformer architectures (Dosovitskiy et al., 2020; Peebles & Xie, 2023) with a large patch size are capable of accurate pixel-space generation when adjusting timestep sampling during training and predicting clean visual information. They argue that under the manifold assumption clean data has lower intrinsic dimensionality than noise or noise-containing velocities, thus making clean data easier to compress by patchification. Similarly, HDiT (Crowson et al., 2024) presents a hierarchical transformer architecture that substantially reduces computational requirements in pixel-space modeling while maintaining the strong scalability of vision transformers. However, due to the additional temporal dimension, pixel-space generation of video or weather forecasting data is especially challenging and computationally demanding (Jabri et al., 2023). Closest to our work, FREUD (Schusterbauer et al., 2026b) uses a pixel-space HDiT to reconstruct precipitation fields from compressed latents. They show that a generative decoder modeling the full video sequence increases temporal consistency. However, the FREUD decoder receives strong framewise conditioning information and therefore operates efficiently with low model capacity and few-step inference. Instead, JWS learns end-to-end weather nowcasting directly in pixel-space. We use standard transformer architectures and denoising procedures, maintaining scalability and simplicity.

D.3 Few-Step Sampling

Precipitation nowcasting requires rapid updates (Shi et al., 2015; Ravuri et al., 2021), which is at odds with the iterative nature of generative diffusion and flow models that tend to improve with more function evaluations. Reducing computational complexity and latency by reducing the number of sampling steps required to produce high-quality samples has been a focus of many researchers in recent years. Most prominently, researchers proposed methods to reduce the number of inference steps by modification of the denoising trajectory. DDIM (Song et al., 2020) defines an ODE path for trained diffusion models that can be traversed with larger steps. While Flow Matching methods (Lipman et al., 2023) explicitly define a straighter trajectory during training, which allows faster traversal (Liu et al., 2022). However, these methods typically still require about 20 network evaluations to produce high quality samples.

Few-step distillation approaches first train a default diffusion model and then distill the model's output into a few-step generative model using progressive MSE-based distillation (Salimans & Ho, 2022), adversarial objectives (Sauer et al., 2023), consistency objectives (Song et al., 2023), or distribution matching objectives (Yin et al., 2024b;a). However, distillation introduces additional training complexity by requiring a two-stage process. Additionally, previous research shows distillation often reduces the diversity of samples (Yin et al., 2024a; Gandikota & Bau, 2025), limiting applicability in domains where capturing the full distribution of potential outcomes is essential.

Consistency models (Song et al., 2023) can be used for distillation as well as for training from scratch. Their training objective puts them in the family of Flow Map models (Boffi et al., 2025a) which are trained to directly integrate the denoising trajectory between two diffusion timesteps. Flow maps can be grouped into three categories following Boffi et al. (2025b): First, Eulerian Flow Maps enforce that a change in the starting point should not change the outcome at the destination time. Second, progressive Flow Maps train models s.t. one large jump agrees with the composition of two smaller jumps. Finally, Lagrangian Flow Maps enforce that a change in the destination time corresponds to following the velocity field at the original destination.

MeanFlows (Geng et al., 2025) are the most prominent example of Eulerian Flow Maps, yet require an expensive Jacobian-vector product (JVP) operation to obtain the training target. While improved MeanFlows (Geng et al., 2026) stabilizes the training objective and PixelMeanFlow (Lu et al., 2026) combines improved MeanFlow with JiT (Li & He, 2025), they retain the expensive JVP computation. In comparison SplitMeanFlow (Guo et al., 2025) eliminates the JVP computation and resembles Shortcut model (Frans et al., 2025) training. These methods fall in the category of progressive Flow Maps. Yet, they require multiple network evaluations during training to obtain self-bootstrapped targets, instead of training for superior distributional coverage. Lagrangian Self-Distillation (LSD) (Boffi et al., 2025b) is an example of the final Flow Map category. These approaches still require time differentiation through the network and additional velocity evaluation at the predicted endpoint to derive their target. Moreover, although Flow Map methods enable few-step generation, there is compounding evidence (Huang et al., 2026; Nguyen et al., 2025a) that distributional coverage is sensitive to the training formulation, potentially increasing the difficulty of applying Flow Maps in domains where uncertainty is essential.

Distribution-level objectives offer a promising alternative. Inductive Moment Matching (IMM) (Zhou et al., 2025) constructs a Flow Map by learning mappings between noise levels through matching output distributions instead of individual samples with a maximum mean discrepancy objective. Improving simplicity, Distributional Diffusion Models (DDM) (De Bortoli et al., 2025) show that training a diffusion model to predict a distribution of outputs rather than the conditional mean by supervision with a proper scoring rule suffices to build strong few step generative models. Distributional Diffusion Models are especially promising for weather forecasting applications as they directly train a generative model to produce an ensemble of outputs. However, DDMs require multiple passes through the full network to produce an ensemble of outputs. Yet, improved Distributional Diffusion Models (iDDM) (Anonymous, 2026) show that, combined with strong conditioning on auxiliary noise, late expansion of the ensemble for only a few final layers is sufficient to train a DDM. Additionally, they show that using a time-dependent interpolation between distributional and MSE objectives produces a model that performs well in few and many step inference.

PixelFlowCast (Zhu et al., 2026) built a pixel-space nowcasting method with a (Pixel-)MeanFlow objective (Lu et al., 2026; Geng et al., 2025). However, it only models the residual of a deterministic

forecast. Further, due to the cost of MeanFlow training, their model was only trained for downsampled data. We argue that Distributional Diffusion (De Bortoli et al., 2025; Anonymous, 2026) is a natural fit for weather nowcasting and makes efficient nowcasting with standard transformer architectures (Li & He, 2025; Dosovitskiy et al., 2020; Crowson et al., 2024; Peebles & Xie, 2023; Ma et al., 2024) feasible without biasing generation (Schusterbauer et al., 2026b).

E Qualitative Results

E.1 Qualitative Forecast Ensembles

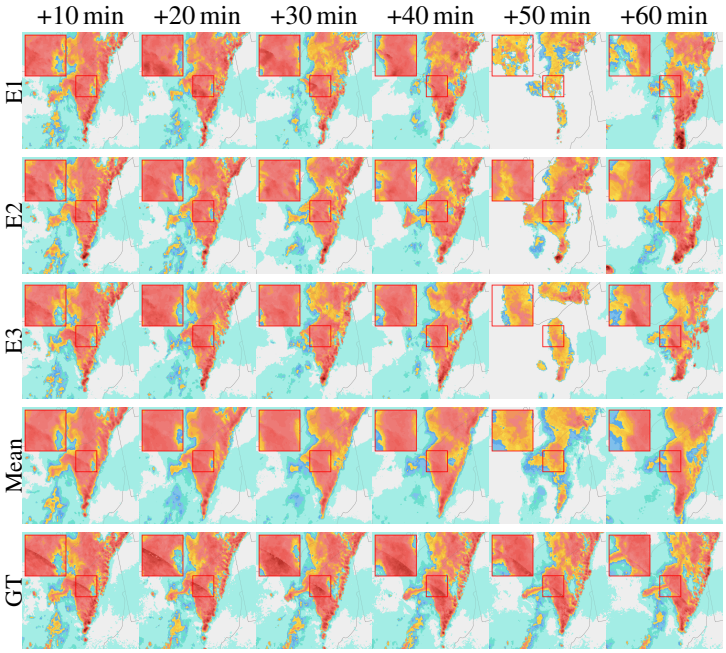


Figure 25: **Qualitative Forecast.** We show three ensemble members over the full 60 min lead time range. Sample from the SEVIR dataset. Zoom shows a high-precipitation region with high variability.

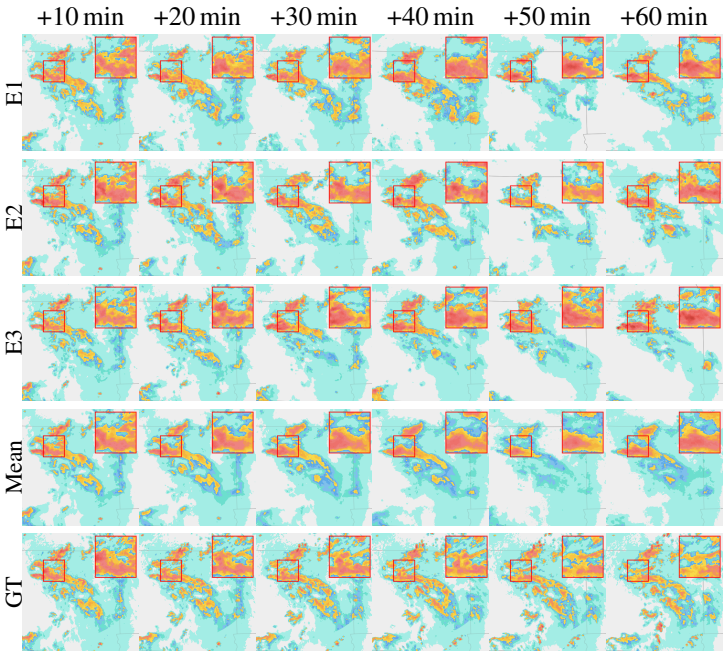


Figure 26: **Qualitative Forecast.** We show three ensemble members over the full 60 min lead time range. Sample from the SEVIR dataset. Zoom shows a high-precipitation region with high variability.

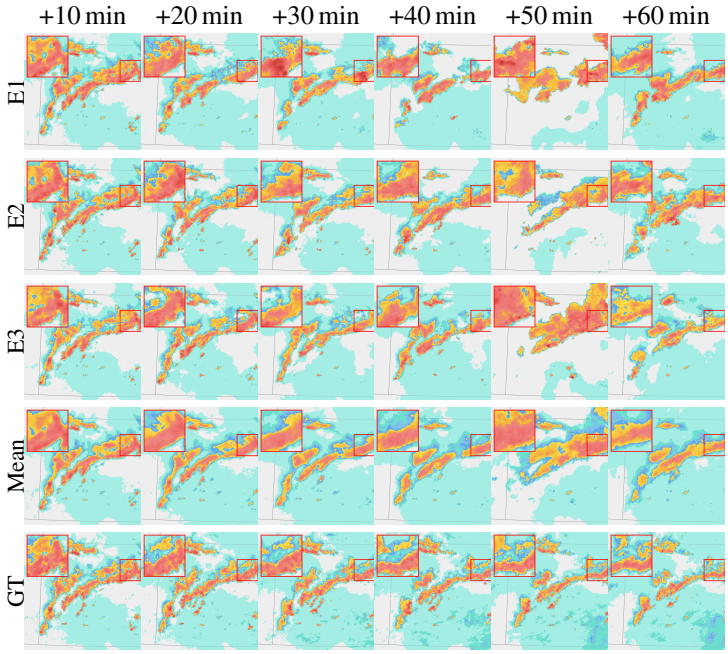


Figure 27: **Qualitative Forecast.** We show three ensemble members over the full 60 min lead time range. Sample from the SEVIR dataset. Zoom shows a high-precipitation region with high variability.

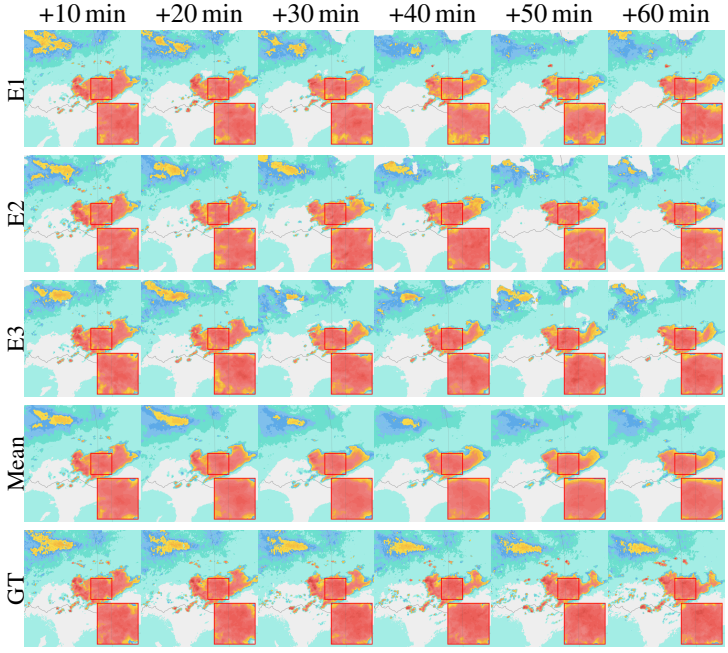


Figure 28: **Qualitative Forecast.** We show three ensemble members over the full 60 min lead time range. Sample from the SEVIR dataset. Zoom shows a high-precipitation region with high variability.

E.2 Qualitative Comparisons

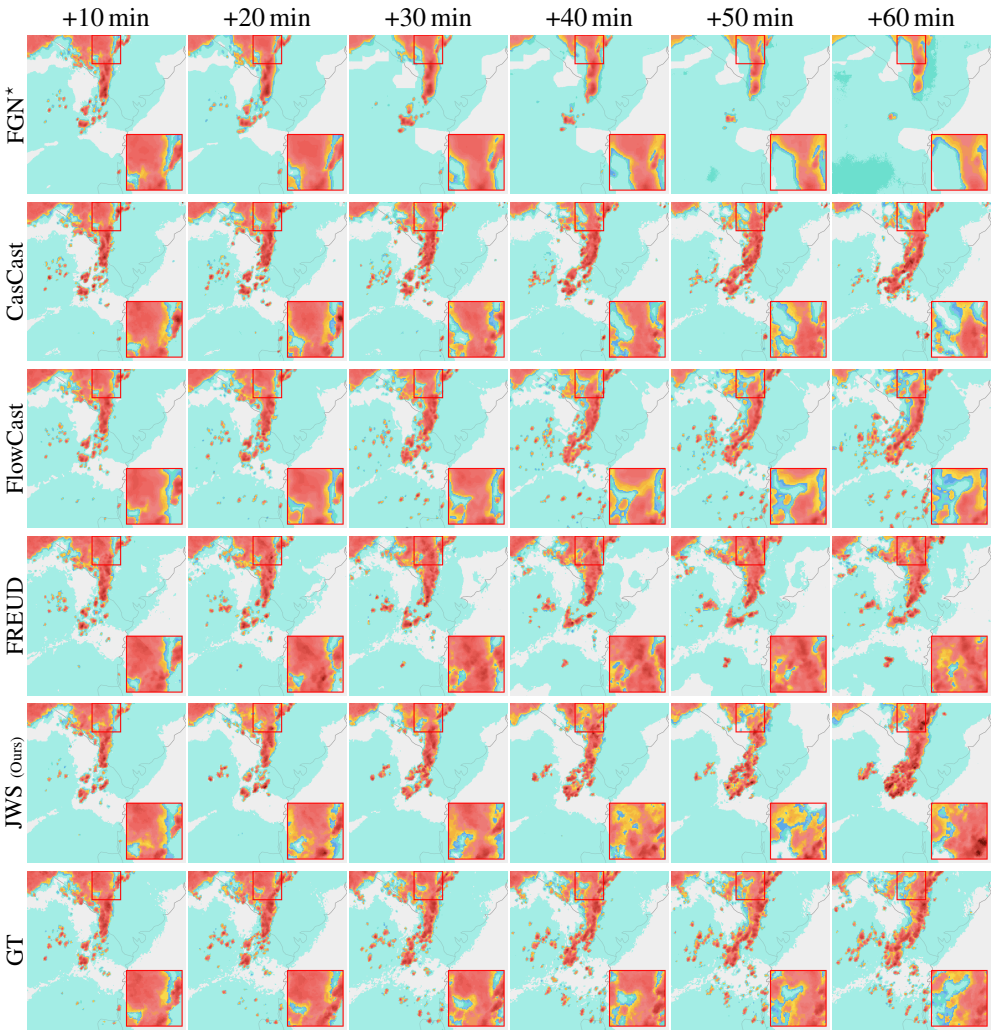


Figure 29: **Qualitative Comparison.** Compared to FGN (Alet et al., 2025), CasCast (Gong et al., 2024), FlowCast (Ribeiro & Pucer, 2026), and FREUD (Schusterbauer et al., 2026b) JWS better captures local and global precipitation patterns. ⋆ model trained by us.

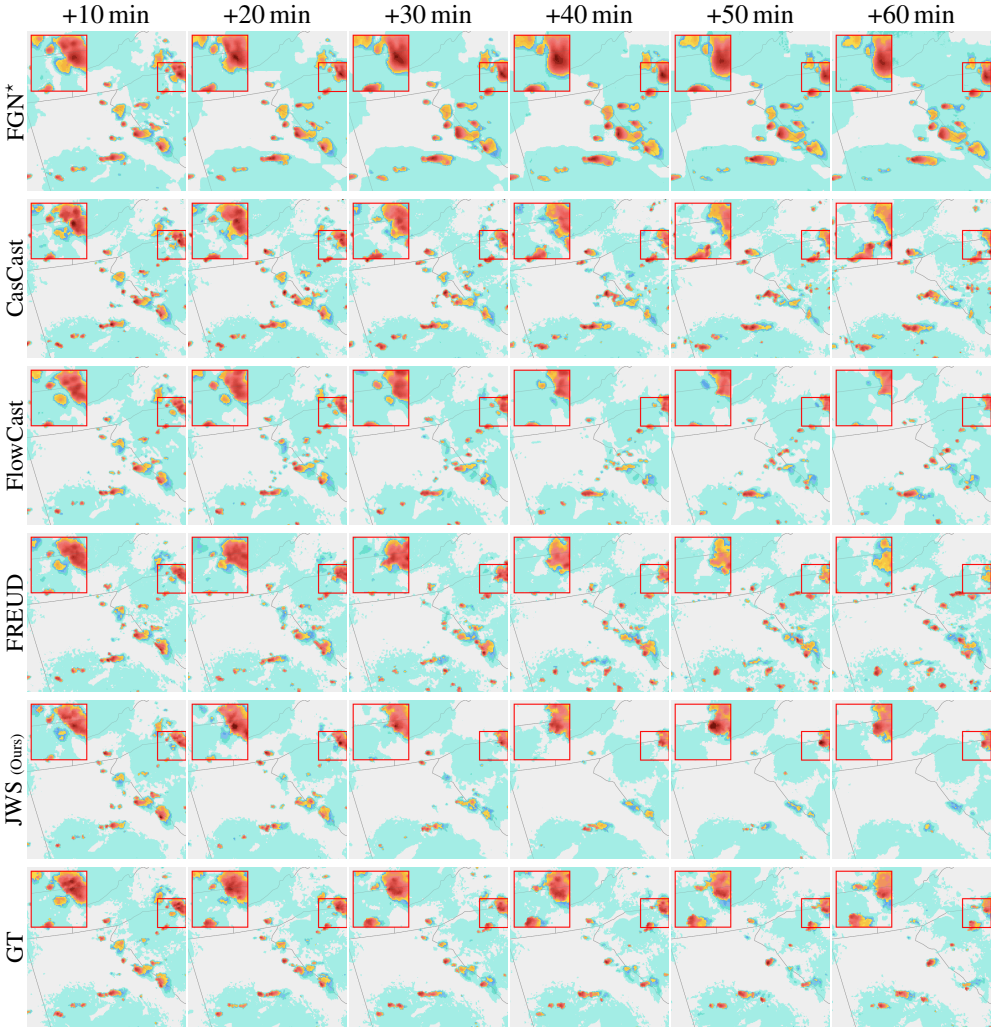


Figure 30: **Qualitative Comparison.** FlowCast (Ribeiro & Pucer, 2026) and JWS capture localized high precipitation accurately. ⋆ model trained by us.

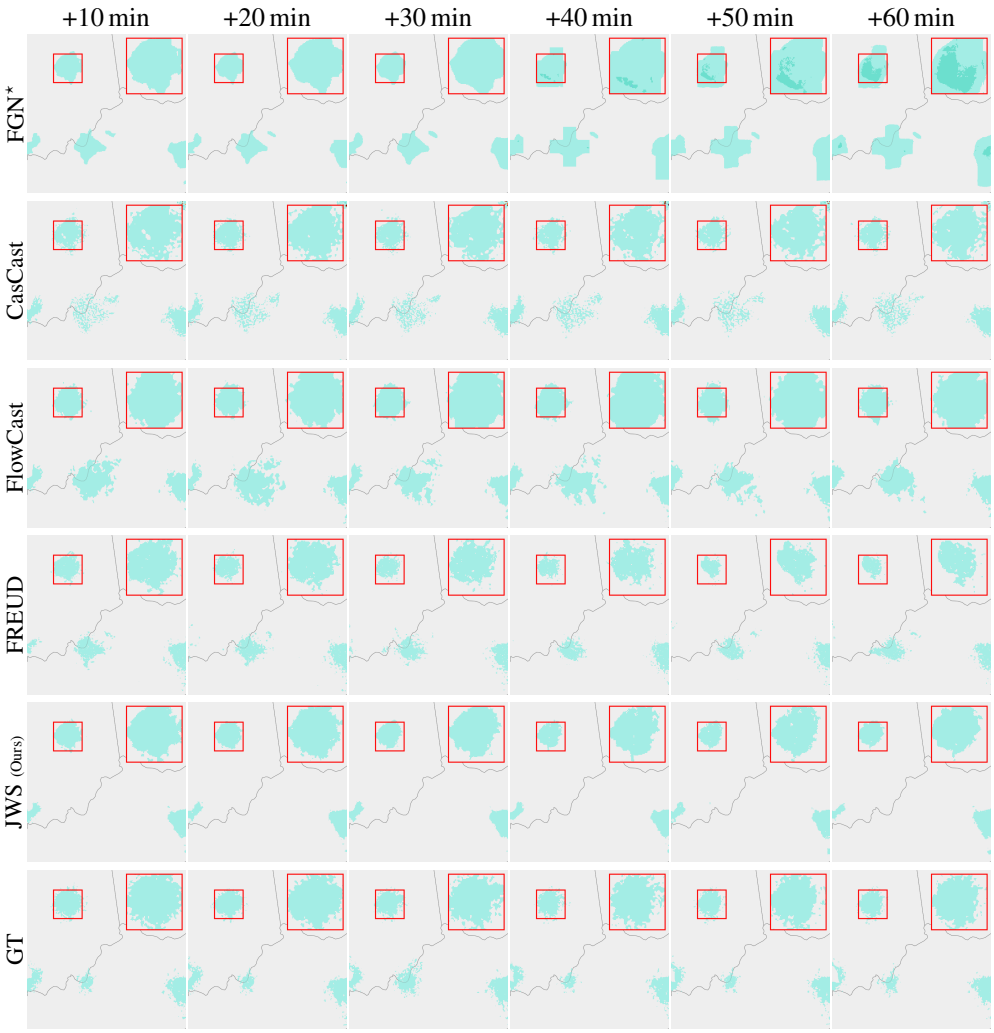


Figure 31: **Qualitative Comparison.** For low precipitation JWS is able to capture dynamics more accurately than FGN (Alet et al., 2025), CasCast (Gong et al., 2024), FlowCast (Ribeiro & Pucer, 2026), or FREUD (Schusterbauer et al., 2026b). ⋆ model trained by us.

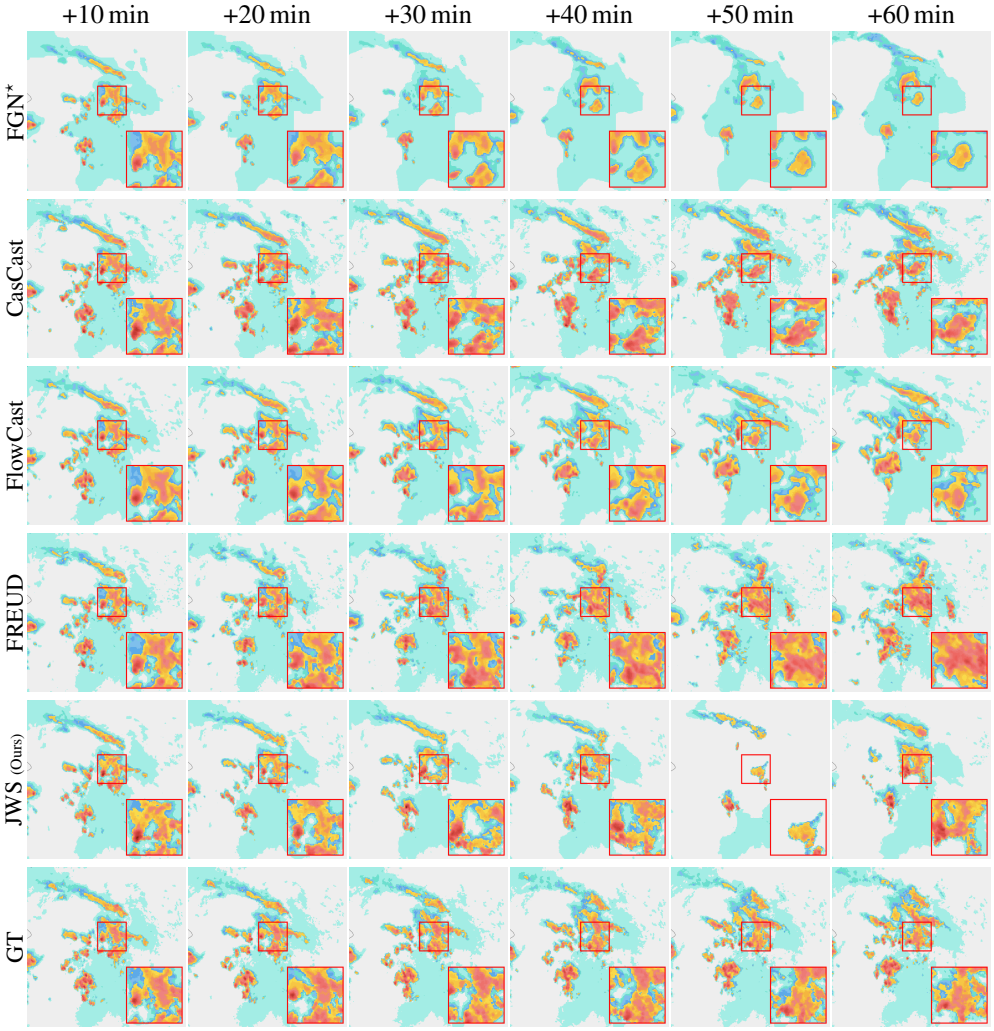


Figure 32: **Qualitative Comparison.** JWS produces sharper and more intricate precipitation patterns than prior work. $\star$ model trained by us.

E.3 Qualitative Uncertainty

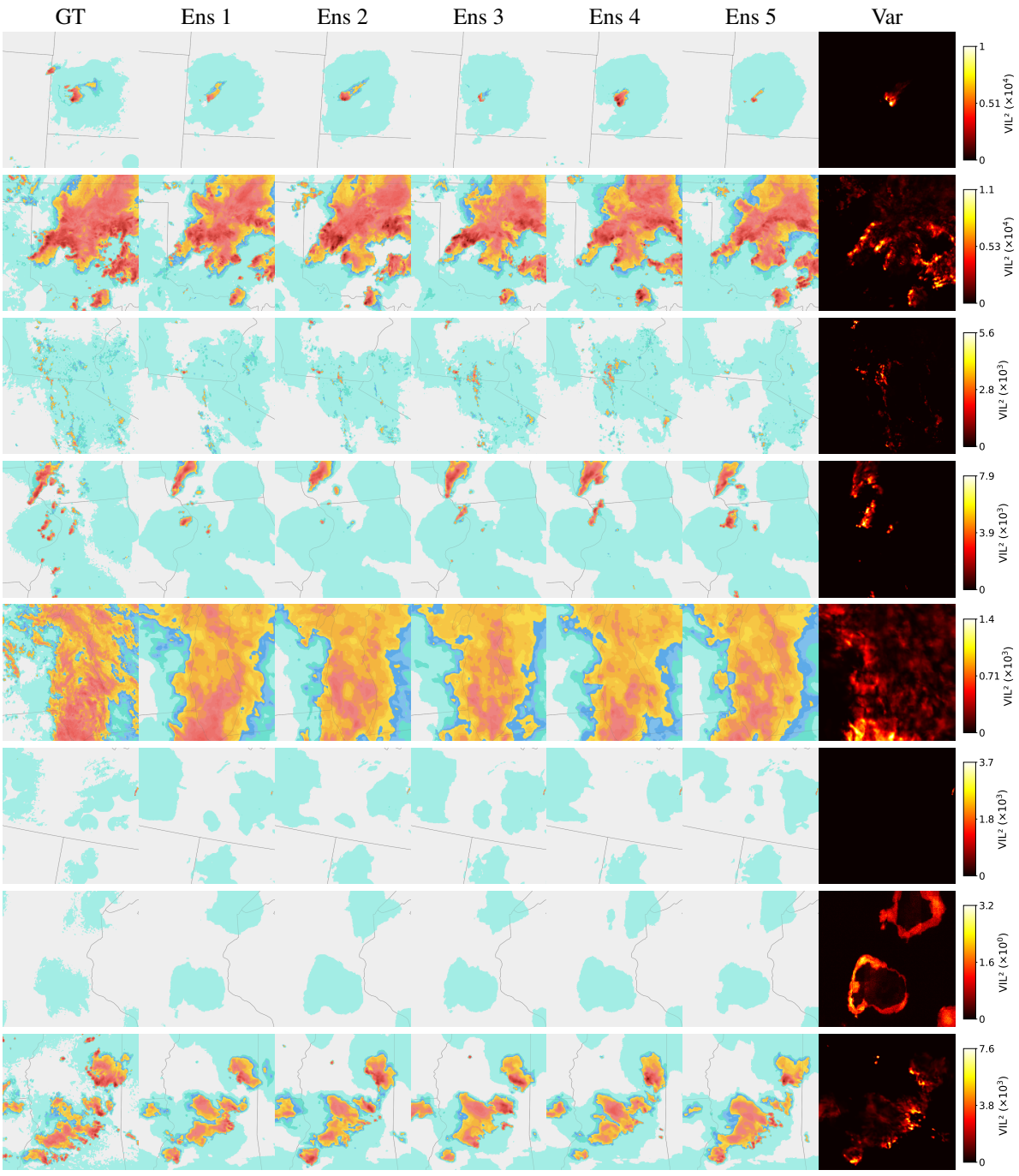


Figure 33: **Ensemble Variance.** We show five ensemble members for samples from the SEVIR dataset at +60 min with complex precipitation patterns. Variance maps highlight localized uncertainty in complex precipitation regions.