NUCLECAST: LOW-COST RESIDUAL FLOW-MATCHING REFINEMENT FOR RADAR NOWCASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Radar nowcasting requires predicting rapidly evolving precipitation fields while accounting for uncertainty in their future development. Deterministic forecasts often smooth over multiple plausible outcomes, whereas generative models represent alternative futures through ensembles. Repeated network evaluations make diffusion and flow-based ensemble sampling computationally and infrastructure-intensive. We introduce NucleCast, a two-stage probabilistic nowcasting framework that refines a deterministic forecast anchor rather than generating the entire future field from noise. An efficient Mamba-3-based forecaster first predicts the large-scale evolution. A pixel-space conditional flow matching model then directly predicts clean residual corrections using a mask-conditioned compound-Poisson source that proposes localized precipitation variations. Each ensemble member requires only four flow matching model evaluations, with the same deterministic anchor shared across members. Evaluated on forecast skill, spatial and spectral structure, NucleCast achieves performance competitive with state-of-the-art generative baselines while drastically reducing computational cost. Furthermore, we investigate newly emerging precipitation by explicitly distinguishing recovered events from false off-support predictions. Our sampling analysis reveals that finer numerical integration generates additional fine-scale structures and off-support precipitation.

1 INTRODUCTION

Precipitation nowcasting is the prediction of the evolution of radar fields over the next few hours to support flood warnings and short-term decision-making. Yet precipitation is inherently uncertain: convective cells can intensify, decay, split, or emerge in multiple plausible ways. Deterministic forecasts tend to smooth out these possible outcomes, while generative models address this limitation by sampling multiple coherent forecasts, providing an ensemble representation of forecast uncertainty (Ravuri et al., 2021). While diffusion-based approaches have improved forecast realism, their iterative sampling remains computationally demanding (Gao et al., 2023; Yu et al., 2024).

Conditional Flow Matching (CFM) provides an attractive alternative by learning continuous transport from a source distribution to the forecast distribution (Gagneux et al., 2025). FlowCast demonstrated its effectiveness for precipitation nowcasting through noise-to-data transport in a compressed latent space (Perrone Ribeiro & Faganeli Pucer, 2026), which requires a separately trained autoencoder. Building on findings that directly predicting clean data enables effective pixel-space generation (Li & He, 2026), PixelFlowCast (Zhu et al., 2026) explored deterministic forecast guidance within this x-prediction framework. This raises a practical question: *in rapidly evolving radar sequences, why generate every future frame from noise when much of the large-scale motion can already be predicted deterministically?*

This question becomes particularly important under realistic compute constraints. Spatiotemporal forecasters such as SimVP, Earthformer, and CasCast rely on convolutional or attention-based architectures (Tan et al., 2025; Gao et al., 2022; Gong et al., 2024); repeatedly training large deterministic and generative components can make probabilistic nowcasting difficult to reproduce and deploy outside large accelerator clusters. State-space models offer a complementary efficiency path. Mamba (Gu & Dao, 2023) introduced selective state-space modeling with linear sequence-length scaling, and STVMamba (Zou et al., 2025) sucesfully adapted this family to precipitation nowcasting. More

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.

recently, Mamba-3 advanced the performance–efficiency trade-off through richer state updates and multi-input, multi-output formulations (Lahoti et al., 2026).

We introduce **NucleCast**, a two-stage probabilistic nowcasting system that learns to refine rather than regenerate. Our efficient deterministic forecaster is a STVMamba with Mamba-3 blocks and six directional space–time scans that are able to capture predictable large-scale evolution. A lightweight conditional flow model predicts clean residual corrections. This decomposition assigns the generative stage to unresolved precipitation variability, including local intensification and fine-scale structural changes, without requiring it to relearn the full forecast.

To represent plausible residual variations, NucleCast combines a structured compound-Poisson source with conditional flow matching in pixel space. The model generates an ensemble of plausible futures following the probabilistic forecasting perspective of Ravuri et al. (2021). We use an eight-member probability-matched mean (PM-mean) (Ebert, 2001) for deterministic verification, while retaining individual ensemble members to represent forecast uncertainty. The efficient state-space backbone and low-cost residual sampling target produce probabilistic nowcasting under limited computational resources.

Our contributions are summarized as follows:

- **Efficient residual flow matching for probabilistic nowcasting.** We combine a Mamba-3-based deterministic forecaster with conditional flow matching over residual corrections. The deployed refiner requires four flow evaluations per ensemble member, providing a low-cost route to probabilistic radar forecasts.
- **Multimodal, high-frequency refinement around a deterministic forecast.** By delegating predictable large-scale evolution to the deterministic stage, NucleCast models unresolved precipitation variability through residual flow matching using a structured compound-Poisson source to propose localised corrections and sample multiple plausible fine-scale evolutions.
- **A compute-aware evaluation of stochastic refinement.** We evaluate forecast skill, extreme-event localisation, and inference cost against generative baselines. We further analyse rare-event recovery and show that finer numerical integration can increase off-support generation without a detectable improvement in localisation at equal escape budget.

2 Related Work

Generative nowcasting and deterministic–stochastic cascades. Generative radar nowcasting represents uncertainty through alternative future realizations (Ravuri et al., 2021). PreDiff (Gao et al., 2023) applies latent diffusion to precipitation sequences, while CasCast (Gong et al., 2024) and DiffCast (Yu et al., 2024) combine deterministic prediction with generative modeling . DiffCast explicitly models the residual around a deterministic forecast using diffusion; residual generation itself is therefore not new to our work. RectiCast further distinguishes systematic mean-field shift from stochastic variability through a rectifier followed by a generator (Ju et al., 2026). NucleCast instead couples a frozen state-space forecaster with clean-residual flow matching and a structured conditional source, targeting few-evaluation ensemble inference.

Flow matching and clean-data prediction. FlowCast (Perrone Ribeiro & Faganeli Pucer, 2026) applies conditional flow matching to entire future sequences in a compressed latent space, supported by a separately trained autoencoder. Li & He (2026) show that directly predicting clean images can enable effective high-dimensional pixel-space generation; PixelFlowCast explores pixel-space clean prediction with deterministic guidance for precipitation nowcasting (Zhu et al., 2026). These works motivate clean prediction but do not establish that its natural-image behavior transfers unchanged to conditional radar sequences. In particular, subtracting a deterministic forecast translates the conditional target distribution without removing its uncertainty (Appendix A.1.5). We predict the *clean residual*, not the complete future field, and recover the flow velocity analytically (Appendix A.1.1).

Efficient spatiotemporal forecasting. SimVPv2 provides a convolutional forecasting baseline, whereas Earthformer models space–time dependencies with attention (Tan et al., 2025; Gao et al.,

2022). Mamba offers selective state-space sequence modeling (Gu & Dao, 2023); STVMamba adapts directional scans to precipitation nowcasting (Zou et al., 2025), and Mamba-3 advances the sequence-modeling performance–efficiency trade-off (Lahoti et al., 2026). We adapt this line into the deterministic STVMamba3 anchor and examine its architectural choices separately in an appendix.

3 METHOD

NucleCast couples a frozen deterministic nowcaster with conditional flow matching refiner over its forecast residual. Let x_{past} denote the twelve observed radar frames and Y the future sequence. Unless specified otherwise, radar fields and flow states in the equations use standardized Vertically Integrated Liquid (VIL) coordinates. The deterministic anchor is denoted by D, and the refiner's full conditioning context is $C = (x_{\text{past}}, D, M)$, where M is an anchor-derived mask. Thus x_{past} is the input to the deterministic forecaster, whereas C conditions both the residual source and the flow network. Figure 1 summarizes the system and separates inference from training-only supervision.

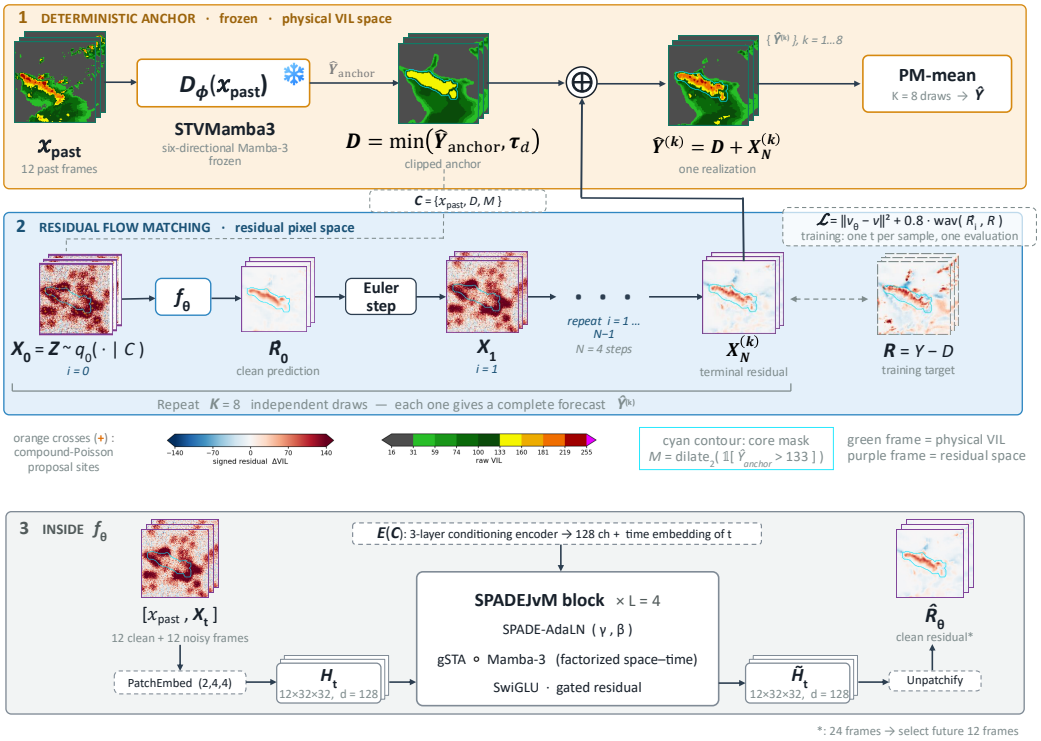


Figure 1: **Overview of NucleCast. (1)** A frozen STVMamba3 anchor D is generated from x_{past}. Conditioned on $C = (x_{\text{past}}, D, M)$, **(2)** residual flow matching integrates a structured source $X_0 = Z \sim q_0(\cdot \mid C)$ to X_N in four Euler steps, yielding forecasts $\widehat{Y}^{(k)} = D + X_N^{(k)}$. Eight draws are aggregated via PM-mean. **(3)** Inside f_θ, concatenated inputs $[x_{\text{past}}, X_t]$ are patch-embedded, processed through four SPADE–Mamba blocks conditioned on (C, t), and unpatchified to predict the clean residual $\widehat{R}_t$ (Appendix Fig. 10).

3.1 DETERMINISTIC FORECASTING

Our first stage, **STVMamba3**, maps x_{past} to 12 future frames. It builds on STVMamba (Zou et al., 2025), comphrehensibly described in Appendix A.4 The backbone is trained with a deterministic squared-error objective and frozen before fitting the refiner.

Let

$$\widehat{Y}_{\text{anchor}} = D_\phi(x_{\text{past}}) \tag{1}$$

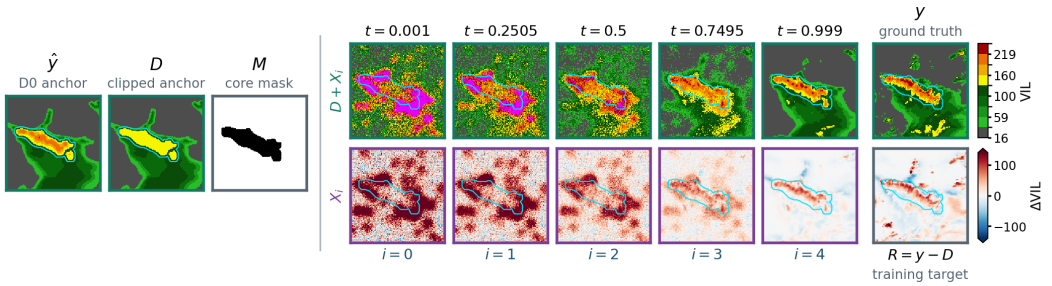


Figure 2: **NucleCast inference trajectory (window 122, lead 10/12). Left:** Deterministic anchor $\widehat{y}$, clipped anchor $D = D(x_{\text{past}})$, and dilated core mask M (conditioning only). **Right:** Four-step Euler rollout ($t \in [0.001, 0.999]$). *Top:* Physical VIL states ($D + X_i$). *Bottom:* ODE residual states X_i, terminating near $R = y - D$. Over the rollout, the refiner removes proposals outside the storm and sharpens the core toward ground truth y.

denote the deterministic forecast in physical radar coordinates. NucleCast uses a dataset-specific core threshold τ_d to define both the residual anchor and its conditioning mask:

$$D = \mathcal{S}_d\Big(\min\Big(\widehat{Y}_{\text{anchor}}, \tau_d\Big)\Big), \tag{2}$$

$$M = \mathcal{D}_2\Big(\mathbf{1}[\widehat{Y}_{\text{anchor}} > \tau_d]\Big), \tag{3}$$

where $\mathcal{S}_d$ denotes the fixed dataset-specific standardization and $\mathcal{D}_2$ a two-pixel dilation (details in Appendix A.1.1). We use $\tau_{\text{SEVIR}} = 133$ VIL and $\tau_{\text{ARSO}} = 30$ dBZ. The complete conditioning context is

$$C = (x_{\text{past}}, D, M). \tag{4}$$

Writing Y in the same standardized coordinates as D, the second-stage target is

$$R = Y - D. \tag{5}$$

The physical VIL maps in Figure 1 display the inverse transform $y_{\text{VIL}} = 33.44 + 47.54y$, whereas residual maps display signed $\Delta\text{VIL} = 47.54r$ without adding the mean. Architectural diagrams, parameter accounting, and ablations against SimVP-gSTA and a four-direction variant are provided in Appendix A.4. The effect of anchor clipping is explained is A.1.6. Residualization is a conditional translation and does not itself remove forecast uncertainty (Appendix A.1.5).

3.2 Clean-Residual Flow Matching

During training, we draw source proposals $Z \sim q_0(\cdot \mid C)$ independently of the residual target R, and sample time steps from a logit-normal schedule, $t = \sigma(\varepsilon)$ with $\varepsilon \sim \mathcal{N}(-1.5, 1)$. We then construct the probability path and target velocity:

$$X_t = (1 - t)Z + tR, \qquad u_t = R - Z. \tag{6}$$

The network is parameterized to predict the clean residual rather than velocity directly:

$$\widehat{R}_t = f_\theta(X_t, t, C), \qquad v_\theta(X_t, t, C) = \frac{\widehat{R}_t - X_t}{1 - t}. \tag{7}$$

We optimize the velocity matching loss augmented with a wavelet penalty:

$$\mathcal{L} = \mathbb{E}_{Z,R,t}\left[\|v_\theta(X_t, t, C) - u_t\|_2^2 + \lambda_{\text{wav}}\mathcal{L}_{\text{wav}}(\widehat{R}_t, R)\right]. \tag{8}$$

where $\mathcal{L}_{\text{wav}}$ computes a subband-weighted L_1 distance on 2D wavelet detail coefficients (defined in Appendix A.2), with $\lambda_{\text{wav}} = 0.8$ fixed across all experiments.

Each training sample requires a single evaluation of f_θ; the iterative Euler sampler used at inference is described in Section 4.1.4.

3.3 Structured Source and Few-Step Sampling

The source $q_0(Z \mid C)$ superimposes white Gaussian noise with positive, Gaussian-shaped compound-Poisson proposals. Proposals are sampled within an annular region around the deterministic core support, defined by thresholding the unclipped anchor at raw VIL 133 followed by a two-pixel dilation. This mask acts as a conditioning channel and proposal guide, but is never enforced as a hard output constraint. Figure 11 compares Gaussian noise source and NucleCast's source in Appendix A.6.

At inference, each ensemble member initializes at $X_0 \sim q_0(\cdot \mid C)$ (member index omitted for clarity). On a uniform grid with $N = 4$ steps ($t_i = i/N$, $\Delta t = 1/N$), each step proceeds as:

$$\underbrace{\widehat{R}_i = f_\theta(X_i, t_i, C)}_{\text{Clean prediction}} \longrightarrow \underbrace{v_i = \frac{\widehat{R}_i - X_i}{1 - t_i}}_{\text{Velocity}} \longrightarrow \underbrace{X_{i+1} = X_i + \Delta t\, v_i}_{\text{Euler step}}. \tag{9}$$

The four-step discretization requires exactly **four refiner evaluations per member** (none at $t_N = 1$). Integration operates directly in standardized residual pixel space, avoiding any auxiliary latent autoencoder. The final forecast is recovered as $\widehat{Y} = D + X_N$, mapped to physical units via $\mathrm{clip}_{[0,255]}\left(33.44 + 47.54\widehat{Y}\right)$.

3.4 Ensembles and Longer-Horizon Feedback

Drawing K independent sources yields an ensemble $\{\widehat{Y}^{(k)}\}_{k=1}^{K}$ around the deterministic anchor D. We set $K = 8$ (32 refiner evaluations per window) and aggregate members via a PM-mean (Ebert, 2001), mapping the spatial ranks of the ensemble mean to the pooled member intensities. Individual members are retained for probabilistic verification.

For longer horizons, the model rolls out autoregressively in one-hour blocks. In *feedback* mode, the PM-mean serves as the input context for subsequent blocks. In *output-only* mode, the coarse anchor drives the rollout, applying refinement only to the final reported frames. A *gated* policy switches between coarse and refined feedback at a chosen lead time (details in Appendix A.8).

4 Experiments

4.1 Experimental Setting

4.1.1 Datasets

We validate our approach using two distinct 1 km resolution radar datasets: **SEVIR** (Veillette et al., 2020) for standard US continental benchmarking, and **ARSO**, an operational Slovenian radar composite to assess regional generalization. In line with standard nowcasting formulations (Shi et al., 2015), the model forecasts the subsequent 12 future frames (60 minutes) conditioned on an observed history of 12 frames (60 minutes) on SEVIR and 13 frames (65 minutes) on ARSO, following FlowCast's respective protocol.

Table 1: Radar datasets, grid configurations, and chronological partitions.

Dataset	Native Grid	Model Grid	N_{train}	N_{val}	N_{test}	Frames (In/Out)
SEVIR	384×384	128×128	36,351	9,450	12,420	$12 \to 12$ (+60 min)
ARSO	301×401	$304 \times 416^{\dagger}$	30,958	10,303	10,398	$13 \to 12$ (+60 min)

†Replication-padded for patch divisibility; cropped back to 301×401 for metric evaluation.

SEVIR. SEVIR (Veillette et al., 2020) provides radar-derived Vertically Integrated Liquid (VIL) mosaics over the continental United States at 1 km native resolution and 5-minute intervals (see Table 1). In line with our objective of lightweight and accessible probabilistic nowcasting, inputs and targets are downsampled via 3×3 average pooling to 128×128 (3 km/pixel); all SEVIR evaluations are reported at this resolution. To ensure a fair comparison between FlowCast, DiffCast, and NucleCast, we evaluate on Cohort B: 184 timestamps (4,332 sequences, representing 34.9% of the test archive) forming the common test split shared with DiffCast.

ARSO. ARSO provides 1 km, 5-minute radar reflectivity composites over Slovenia (Table 1). We follow FlowCast's 60/20/20 chronological split and retain the native 301×401 spatial resolution, both to test NucleCast at substantially larger spatial scale and to remain as close as possible to the published FlowCast evaluation protocol. We use the same $12 \rightarrow 12$ forecasting setup as on SEVIR and evaluate forecasts directly in dBZ using FlowCast's thresholds $\{15, 21, 30, 33, 36, 39\}$.

4.1.2 Evaluation Metrics

All evaluations are conducted in physical data units on the 12 forecast leads. Predictions are evaluated as single draws ($N = 1$) or 8-member ensembles aggregated via probability-matched mean (PM8).

Continuous and Categorical Skill.

- **CRPS:** Evaluates probabilistic calibration across all 8 realization members via a fitted Gaussian distribution; collapses to Mean Absolute Error (MAE) when $N = 1$.
- **CSI-M & HSS-M:** Mean Critical Success Index and Heidke Skill Score averaged across all leads and registered thresholds. For SEVIR, $\Theta_{\mathrm{VIL}} = \{16, 74, 133, 160, 181, 219\}$; for ARSO, the quantile-matched set is $\Theta_{\mathrm{ARSO}} = \{15, 21, 30, 33, 36, 39\}$ dBZ.
- **CSI-p8-M:** Neighborhood-pooled CSI computed over non-overlapping 8×8 blocks (24×24 km) to accommodate minor convective phase displacement.
- **FSS$_5$@160:** Fractions Skill Score within a 5×5 window (15 km) at threshold 160 ($>$ 45 dBZ).
- **SSIM & RAPSD$_{\mathrm{hf}}$:** Structural similarity across the full domain, and the spectral power ratio across the five highest radial wavenumber bins ($\lambda \leq 15$ km, target 1.0).

Extreme Precipitation and Area Balance. On extreme thresholds (VIL 160 for storm cores, 219 for severe hailstorms), we report **CSI@160**, **CSI@219**, Probability of Detection (**POD**), and False Alarm Ratio (**FAR**). We jointly assess the exceedance area ratio: $A = \frac{\mathrm{POD}}{1-\mathrm{FAR}} = \frac{N_{\mathrm{pred}}}{N_{\mathrm{obs}}}$, which unmasks whether high detection is gained through calibrated refinement ($A \approx 1.0$) or pathological over-forecasting ($A > 1$).

4.1.3 Training Details

The deterministic STVMamba3 anchor is optimized with MSE loss and frozen. NucleCast refines residual targets $R = Y - D$ using independent source–target pairing, logit-normal time scheduling, and velocity matching augmented with an auxiliary wavelet loss ($\lambda_{\mathrm{wav}} = 0.8$). The refiner velocity network is trained on 4 NVIDIA H100 GPUs for 20 epochs using the AdamW optimizer with a OnceCycle learning rate schedule.

4.1.4 Inference Details

NucleCast computes the deterministic anchor once and generates $K = 8$ stochastic realizations on a 5-point Euler grid corresponding to a four Euler update and NFE $= 4$ per member (32 total evaluations), for more details see Appendix A.3.1. Ensemble forecasts are aggregated via PM-mean. On an NVIDIA A100 GPU, complete 60-minute sequence latency is 0.22 s for a single draw and 0.92 s for the full 8-member PM8 product, including the shared upstream anchor forward pass.

4.2 Comparison to the state of the art

We compare NucleCast with its deterministic STVMamba3 anchor (Zou et al., 2025) and the SimVP-gSTA (Tan et al., 2025) baseline, together with two released probabilistic systems in their native configurations. DiffCast (Yu et al., 2024) is evaluated from its released PhyDNet (Guen & Thome, 2020) checkpoint under its native $5 \rightarrow 20$ protocol, scoring the first 12 forecast frames, while FlowCast (Perrone Ribeiro & Faganeli Pucer, 2026) retains its native $13 \rightarrow 12$ SEVIR conditioning and 384×384 resolution. We used plain 3×3 average (area) pool to downscale its predictions to 128×128. All comparisons therefore use the same forecast targets while preserving each system's released conditioning history.

System	Computational Cost			Error	Forecast Skill						Spatial and Structural Quality		
	Params (M) ↓	NFE /mem ↓	Latency (s) ↓	CRPS ↓	CSI-M ↑	CSI-p8-M ↑	CSI@219 ↑	HSS-M ↑	POD-M ↑	FAR-M ↓	FSS_5@160 ↑	SSIM ↑	$RAPSD_{hf}$ → 1
Single Realisation ($N = 1$)													
STVMamba3 (anchor)	**9.4**	**1**	0.12	**0.0225**	0.3803	0.4184	0.1108	0.4760	0.4524	**0.3858**	0.5191	0.7703	0.3016
SimVP-gSTA (deterministic)	46.8	**1**	**0.01**	0.0234	0.3677	0.4184	0.0959	0.4612	0.4448	0.4369	0.4966	0.7697	0.3227
DiffCast (PhyDNet, 1 draw)	52.1	250	19.15	0.0363	0.2735	0.3869	0.0585	0.3659	0.3499	0.5738	0.4384	0.6735	**1.0443**
NucleCast (ours, 1 draw)	17.2	4	0.22	0.0240	0.3705	0.4860	0.1033	0.4738	0.4653	0.4853	0.5760	0.7697	0.3294
FlowCast (1 draw)	160.5	10	1.76	0.0225	**0.3948**	**0.5280**	**0.1391**	**0.5044**	**0.4878**	0.4461	**0.6312**	**0.7814**	0.5486
Aggregated Operating Point ($N = 8$ Probability-Matched Mean)													
NucleCast (ours, PM8)	**17.2**	**4**	**0.92**	0.0170	0.3940	0.4777	0.1228	0.5008	0.4871	0.4533	0.5919	0.7934	0.3309
FlowCast (PM8)	160.5	10	13.05	**0.0158**	**0.4263**	**0.5234**	**0.1682**	**0.5397**	**0.5169**	**0.4036**	**0.6572**	**0.8084**	**0.3901**

Table 2: Comparison of NucleCast performance and computational cost with baseline models on SEVIR dataset, cohort B. All metrics are computed over a 12-step forecast. Best in **bold**, second-best underlined within each comparable block.

4.2.1 General performance

As shown in Table 2, NucleCast retains competitive forecast skill at substantially lower inference & training cost. A single realization is slightly below the deterministic anchor in CSI-M (-0.0098), while improving neighborhood and spatial scores such as CSI-p8-M ($+0.676$) and FSS_5@160 ($+0.569$). Aggregating eight members raises all categorical metrics compared to single draw, with a complete latency of 0.92 s, but loses in sharpness at scales < 15 km, as does FlowCast ($RAPSD_{hf}$ -21.77%). FlowCast remains stronger on most general metrics, reaching 0.4263 CSI-M with PM8 compared to 0.3940 for NucleCast, but requires 13.05 s and 160.5 M parameters compared with 17.2 M for NucleCast. DiffCast recovers substantially more high-frequency spectral power, but this sharpness does not translate into comparable forecast skill (see Figure 3), with lower CSI-M, FSS and SSIM at a latency of 19.15 s.

4.2.2 Performance on extreme events

Table 3 compares high-intensity precipitation skill on SEVIR and ARSO using the dataset-specific extreme thresholds defined in Section 4.1.2. SEVIR results are evaluated on the 128×128 area-averaged grid used by NucleCast, whereas ARSO is evaluated at its native 301×401 resolution. On SEVIR, STVMamba3 outperforms SimVP-gSTA on all metics. NucleCast PM8 improves over its deterministic anchor at both storm and extreme threshold, increasing CSI@160 by $+0.0229$ and CSI@219 by $+0.0120$, although FlowCast remains stronger at this operating point. On ARSO, NucleCast provides a larger improvement over the anchor: PM8 improves CSI-M by $+0.181$ and CSI@39 by $+0,0437$ compared to STVMamba3. The published FlowCast eight-member ensemble reports a lower CSI-M and a lower CSI@39 compared to NucleCast PM8. However, their model yields fewer false alarms, achieving a lower FAR than our NucleCast ensemble. Conversely, the absence of reported POD on FlowCast paper prevents a direct comparison of the Area Balance ratio. DiffCast reaches an A@219 close to one, indicating that its pooled extreme-area budget is close to that of the observations, yet its high FAR and low CSI show that this area is often misplaced.

System	SEVIR (Cohort B)					ARSO					
	CSI@160 ↑	CSI@219 ↑	POD@219 ↑	FAR@219 ↓	A@219 → 1	CSI-M ↑	CSI@36 ↑	CSI@39 ↑	POD@39 ↑	FAR@39 ↓	A@39 → 1
Single Realisation ($N = 1$)											
STVMamba3 (anchor)	0.1983	0.1108	0.1485	**0.6648**	0.322	0.4130	0.2414	0.1611	0.2371	**0.6536**	0.583
SimVP-gSTA	0.1814	0.0959	0.1265	0.7221	0.310	–	–	–	–	–	–
DiffCast (PhyDNet, 1 draw)	0.1271	0.0585	0.1094	0.8963	**0.970**	–	–	–	–	–	–
NucleCast (ours, 1 draw)	0.1997	0.1033	0.1596	0.7999	0.683	**0.4140**	**0.2512**	**0.1894**	**0.3144**	0.7053	**1.027**
FlowCast (1 draw)	**0.2250**	**0.1391**	**0.1963**	0.7206	0.647	–	–	–	–	–	–
Aggregated Operating Point ($N = 8$)											
NucleCast (ours, PM8)	0.2212	0.1228	0.1792	0.7524	**0.630**	**0.4329**	**0.2710**	**0.2054**	0.3368	0.6838	1.026
FlowCast (PM8)	**0.2606**	**0.1682**	**0.2249**	0.6607	0.606	–	–	–	–	–	–
FlowCast (ens. mean of 8)	0.2397	0.1429	0.1698	**0.5981**	0.350	0.420*	0.254*	0.183*	–	**0.589***	–

Table 3: Comparison of NucleCast on extreme precipitation across datasets SEVIR and ARSO. SimVP-gSTA, DiffCast and our FlowCast runs were not trained on ARSO (–). FlowCast (ens. mean of 8) scores the arithmetic mean of the same eight FlowCast members as the PM8 row, the reduction the FlowCast paper uses; its ARSO values (*) are the FlowCast paper's on the same test split (POD and A not reported), not re-run by us.
Best in **bold**, second-best underlined within each comparable block.

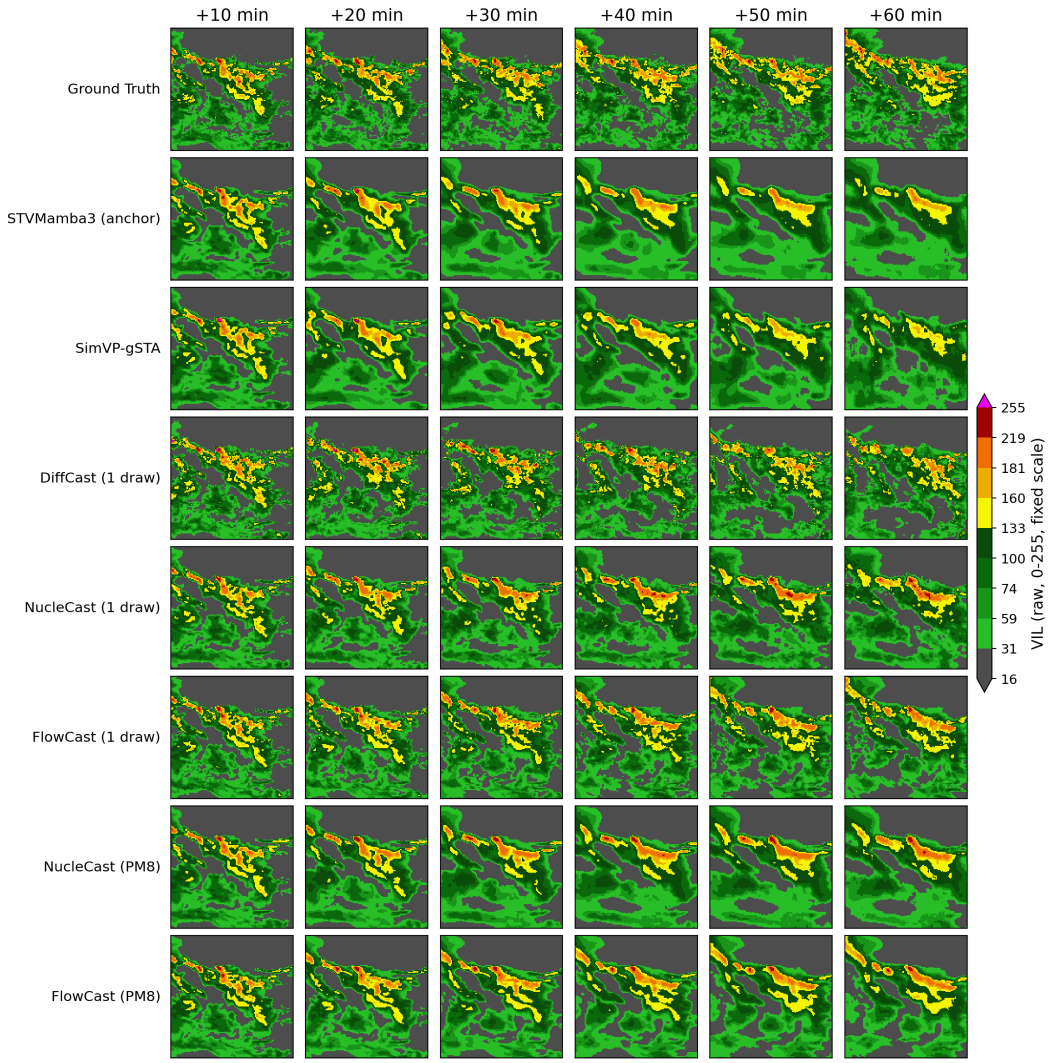


Figure 3: Qualitative comparison of NucleCast with other baselines on a SEVIR sequence. Columns show lead times from 10 to 60 minutes. Rows show the ground-truth, followed by two deterministic models, then generative models. More qualitative comparisons in Appendix A.9

4.3 Ablation and analysis

4.3.1 What does the refiner repair?

For each cascade, we measure the fraction of ground-truth births missed by its deterministic stage that are subsequently recovered by the stochastic refiner, together with newly introduced off-support components. Full definitions and results are reported in Appendix A.7.

NucleCast recovers 25.1% of the birth pixels missed by STVMamba3 in a single draw, while introducing 4,683 new true and 6,759 new false components. PM8 preserves a similar recovery rate (24.9%) while reducing the number of false additions to 4,285, increasing the recovery-to-false ratio from 0.693 to 0.928. Within its own PhyDNet cascade, DiffCast recovers 32.0% of missed birth pixels but introduces 17,642 false components ($\mathrm{RFR} = 0.474$). Since the deterministic anchors miss different events, these recovery rates are interpreted within each cascade rather than as a direct ranking of the refiners.

4.3.2 INFERENCE EFFICIENCY VS. NFE

We assess inference efficiency by varying the number of function evaluations (NFE) used to integrate the same locked NucleCast refiner, where one NFE corresponds to one evaluation of the flow-matching network. Increasing solver accuracy substantially increases inference cost without a corresponding improvement in CSI-M or CSI@219. Although denser integration can increase raw off-support birth recall, its recall at a matched prediction footprint changes little; the full localization results are reported in Table 4 of Appendix A.3. In contrast, using multiple independent members provides a clear skill gain: PM8 improves both CSI-M and CSI@219 over a single NucleCast realization while remaining below one second per forecast.

4.3.3 LONG-HORIZON FEEDBACK

Feeding refined predictions back into the autoregressive rollout improves three-hour skill by $+0.028$ CSI-M over the coarse anchor, whereas output-only refinement fails to sustain skill at extended leads. Full rollout protocols and blockwise breakdowns are detailed in Appendix A.8.

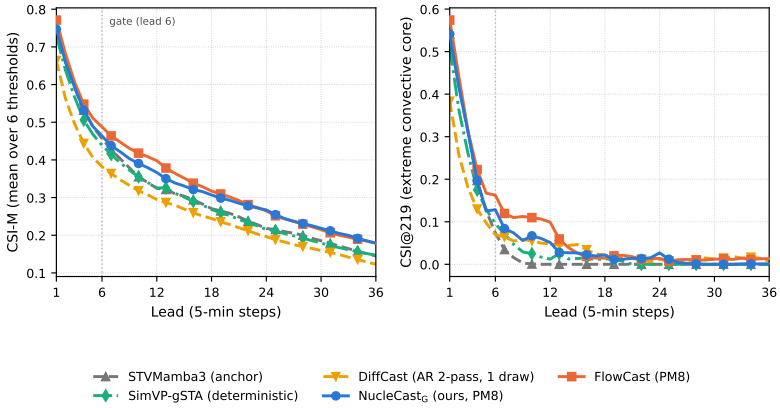


Figure 4: Long-horizon feedback on 96 events from SEVIR test set: CSI-M and CSI@219 vs lead time, +5 to +180 minutes

5 CONCLUSION

We introduced NucleCast, a two-stage probabilistic radar-nowcasting framework that refines a deterministic forecast rather than regenerating the complete future field. NucleCast is designed around a simple trade-off: preserve the predictable large-scale evolution of a deterministic forecast while spending generative capacity only on the remaining residual variability. Across SEVIR and ARSO, NucleCast improves the spatial structure and high-intensity precipitation representation of its deterministic anchor while remaining computationally lightweight. Eight-member aggregation further improves forecast skill at low inference cost, and long-horizon experiments show that feeding refined forecasts back into the deterministic model can preserve these benefits beyond the first prediction window.

The remaining false-alarm increase at high thresholds illustrates the central limitation: generating sharp precipitation structure is easier than placing uncertain future convection correctly. Extreme-event metrics should also be interpreted carefully. On SEVIR, pixels above VIL 219 represent only about 0.052% of the training population, making high-threshold verification particularly sensitive to small spatial displacements. Overall, NucleCast provides a practical route toward probabilistic nowcasting when ensemble quality and computational cost must be considered jointly.

AI USE STATEMENT

In this work we used generative AI tools to **propose and refine hypotheses**, to **design and give feedback on evaluation methodology and experiments**, to **implement methods and evaluation code**, to **clean and reformat datasets** into the evaluation protocols described in Section 3, to **aid in**

polishing sections of the paper. We did not use generative AI tools to generate synthetic datasets, to formulate or prove mathematical claims, to assist with translation, or for qualitative and thematic data analysis; those categories are not applicable to this work. Additionally, we used generative AI tools for tasks in the recommended-disclosure set: creating and editing software code, producing and revising the figures, suggesting sampler and training parameters, searching and summarizing the related literature, drafting parts of this paper, and editing it for readability.

We reviewed all AI-assisted work continuously throughout the research workflow. All reported numbers are systematically verified against our project results ledger via automated consistency checks. Any evaluation code written with AI assistance was cross-validated against reference implementations before computing metrics, and all benchmark comparisons enforce strict hardware consistency. Preliminary hypotheses or numerical claims that failed rigorous verification were discarded during this review process.

Reproducibility Statement

To ensure full reproducibility, our complete codebase, configuration files, and evaluation pipelines are provided in the supplementary material and hosted in an anonymous repository (`https://anonymous.4open.science/r/nuclecast/`). Pre-trained model checkpoints for both the deterministic anchor and the residual refiner are available via an anonymized Hugging Face repository, enabling the direct reproduction of all quantitative tables (Tables 1, 2, and 3) as well as training or fine-tuning models from scratch. Complete specifications of network architectures, loss formulations, and training hyperparameters are documented in Section 3 and in the anonymous repository. Comprehensive descriptions of dataset preparation, padding, sequence generation, and chronological evaluation splits for both SEVIR and ARSO are detailed in Section 4.1.1.

Acknowledgments

We thank the Slovenian Environment Agency (ARSO) for their support and for providing the ARSO dataset used in this work.

References

Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7):909–996, 1988.

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

Anne Gagneux, Ségolène Martin, Rémi Emonet, Quentin Bertrand, and Mathurin Massias. A visual dive into conditional flow matching. In *ICLR Blogposts 2025*, 2025. URL `https://dl.heeere.com/conditional-flow-matching/blog/conditional-flow-matching/`. https://dl.heeere.com/conditional-flow-matching/blog/conditional-flow-matching/.

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, 2022.

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. *Advances in Neural Information Processing Systems*, 36:78621–78656, 2023.

Junchao Gong, Lei Bai, Peng Ye, Wanghan Xu, Na Liu, Jianhua Dai, Xiaokang Yang, and Wanli Ouyang. Cascast: Skillful high-resolution precipitation nowcasting via cascaded modelling. *arXiv preprint arXiv:2402.04290*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

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.

Fanbo Ju, Haiyuan Shi, and Qingjian Ni. Recticast: Rectifying distribution shift in cascaded precipitation nowcasting. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 67–78. Springer, 2026.

Cansu Korkmaz, A Murat Tekalp, and Zafer Dogan. Training generative image super-resolution models by wavelet-domain losses enables better control of artifacts. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5926–5936. IEEE, 2024.

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, volume 2026, pp. 86173–86202, 2026.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Bernardo Perrone Ribeiro and Jana Faganeli Pucer. Flowcast: Advancing precipitation nowcasting with conditional flow matching. In *International Conference on Learning Representations*, volume 2026, pp. 156863–156883, 2026.

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.

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.

Elman Srivastava, Nitish Mansimov and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning (ICML)*, pp. 843–852. PMLR, 2015.

Cheng Tan, Zhangyang Gao, Siyuan Li, and Stan Z Li. Simvpv2: Towards simple yet powerful spatiotemporal predictive learning. *IEEE Transactions on Multimedia*, 27:5170–5184, 2025.

Mark Veillette, Siddharth Samsi, and Chris Mattioli. Sevir: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in neural information processing systems*, 33:22009–22019, 2020.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27758–27767. IEEE, 2024.

Yufeng Zhu, Chunlei Shi, Yongchao Feng, and Dan Niu. Pixelflowcast: Latent-free precipitation nowcasting via pixel mean flows. *arXiv preprint arXiv:2605.10046*, 2026.

Maoyang Zou, Longrui Wen, Yuanyuan Huang, Yuan He, and Jingzhong Xiao. Stvmamba: precipitation nowcasting with spatiotemporal prediction model. *Scientific Reports*, 15(1):22568, 2025.

A Appendix

A.1 Clean Prediction and Residual Flow Matching

This appendix formalizes the clean-prediction parameterization used by NucleCast, relates it to the pixel-space denoising results of Li and He (*Back to Basics: Let Denoising Generative Models Denoise*, CVPR 2026), and clarifies the mathematical role of residualization. The identities below describe an idealized squared-error flow-matching objective. They should not be interpreted as statements about the population optimum of the complete NucleCast training loss, which also contains an auxiliary wavelet-domain term.

A.1.1 CONDITIONAL FLOW MATCHING WITH CLEAN-TARGET PREDICTION

Let x_{past} denote the observed radar history and $Y \in \mathbb{R}^d$ the future sequence in the standardized coordinates used by the flow model. The frozen deterministic forecaster first produces

$$\widehat{Y}_{\mathrm{anchor}} = D_\phi(x_{\mathrm{past}}) \tag{10}$$

in physical radar coordinates.

For dataset d, let τ_d denote the registered core threshold and $\mathcal{S}_d$ the fixed standardization transform. The residual anchor and conditioning mask are

$$D = \mathcal{S}_d\Big(\min(\widehat{Y}_{\mathrm{anchor}}, \tau_d)\Big), \tag{11}$$

$$M = \mathcal{D}_2\Big(\mathbf{1}[\widehat{Y}_{\mathrm{anchor}} > \tau_d]\Big). \tag{12}$$

Here $\mathcal{D}_2$ denotes a two-pixel dilation. The same registered threshold is used for clipping and mask construction: $\tau_{\mathrm{SEVIR}} = 133$ VIL and $\tau_{\mathrm{ARSO}} = 30$ dBZ.

The standardization is affine,

$$\mathcal{S}_d(y) = \frac{y - \mu_d}{\sigma_d}, \tag{13}$$

with dataset-specific statistics (μ_d, σ_d). For SEVIR, $\mu_{\mathrm{SEVIR}} = 33.44$ and $\sigma_{\mathrm{SEVIR}} = 47.54$, so that

$$\mathcal{S}_{\mathrm{SEVIR}}(y) = \frac{y - 33.44}{47.54}, \qquad \mathcal{S}^{-1}_{\mathrm{SEVIR}}(z) = 33.44 + 47.54z. \tag{14}$$

The corresponding ARSO normalization uses the fixed statistics of the ARSO training pipeline.

The complete conditioning variable is

$$C = (x_{\mathrm{past}}, D, M), \tag{15}$$

and the generative stage models the residual

$$R = Y - D. \tag{16}$$

The mask therefore informs where the residual model should focus, without restricting where corrections may be generated.

Let

$$Z \sim q_0(\cdot \mid C) \tag{17}$$

be the conditional source distribution.

For each context $C = c$, consider a joint distribution $\pi_c(z, r)$ whose source and target marginals are respectively $q_0(z \mid c)$ and $p_R(r \mid c)$. The registered NucleCast training procedure uses independent source–target pairing; the following derivation also applies to other couplings.

We define the straight interpolation

$$X_t = (1 - t)Z + tR, \qquad t \in [0, 1]. \tag{18}$$

Differentiating an individual interpolating path gives

$$\dot{X}_t = R - Z. \tag{19}$$

For $0 \le t < 1$, the interpolation also implies

$$Z = \frac{X_t - tR}{1 - t}. \tag{20}$$

Consequently,

$$R - Z = \frac{R - X_t}{1 - t}. \tag{21}$$

Instead of directly predicting the velocity $R - Z$, NucleCast parameterizes its network output as a prediction of the clean residual:

$$\widehat{R}_\theta = f_\theta(X_t, t, C). \tag{22}$$

The corresponding velocity field is recovered algebraically:

$$\boxed{v_\theta(x, t, C) = \frac{f_\theta(x, t, C) - x}{1 - t}, \qquad 0 \le t < 1.} \tag{23}$$

The term involving the current state x is supplied analytically, rather than being reconstructed as part of the network output.

This distinction concerns the quantity represented directly by the network. It does not change the endpoint distribution that the flow is intended to generate.

Population optimum under squared error. Assume that

$$\mathbb{E}\|R\|^2 + \mathbb{E}\|Z\|^2 < \infty. \tag{24}$$

For a fixed time t, consider the clean-target risk

$$\mathcal{L}_t(f) = \mathbb{E}\left[\|f(X_t, t, C) - R\|^2\right]. \tag{25}$$

The L^2 projection identity gives the population minimizer

$$\boxed{f_t^\star(x, c) = \mathbb{E}[R \mid X_t = x, C = c].} \tag{26}$$

Substituting this conditional expectation into equation 23 yields

$$\begin{aligned} v_t^\star(x, c) &= \frac{\mathbb{E}[R \mid X_t = x, C = c] - x}{1 - t} \\ &= \mathbb{E}[R - Z \mid X_t = x, C = c]. \end{aligned} \tag{27}$$

Thus, under the idealized quadratic objective, predicting the clean residual and converting it analytically gives the same population-optimal velocity as direct velocity regression.

This is an equivalence between population-optimal vector fields, not an equivalence between finite neural-network parameterizations. Their optimization problems and approximation errors can differ substantially.

A.1.2 Prediction parameterization versus loss parameterization

Clean prediction and the space in which the loss is evaluated are separate design choices.

Let

$$u = R - Z \tag{28}$$

be the conditional training velocity.

For the exact conversion in equation 23, the following identity holds for every $t < 1$:

$$v_\theta(X_t, t, C) - u = \frac{f_\theta(X_t, t, C) - R}{1 - t}. \tag{29}$$

Therefore,

$$\boxed{\|v_\theta(X_t, t, C) - u\|^2 = \frac{\|f_\theta(X_t, t, C) - R\|^2}{(1 - t)^2}.} \tag{30}$$

A quadratic velocity loss applied to a clean-output network is therefore equivalent to a time-weighted clean-reconstruction loss, provided that the velocity conversion is exactly equation 23.

If training samples t from a density $q(t)$, the corresponding effective weighting of the clean-reconstruction error is

$$w_{\text{eff}}(t) = \frac{q(t)}{(1-t)^2} \tag{31}$$

when the quadratic loss is evaluated in velocity space.

This weighting identity must not be applied unchanged if an implementation modifies the denominator, changes the regression target, or applies nonlinear operations inside the loss.

The actual NucleCast objective additionally contains a wavelet-domain term. Schematically, it consists of a quadratic flow-matching component and auxiliary wavelet supervision:

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{quadratic}} + \lambda_{\text{wav}} \mathcal{L}_{\text{wav}}. \tag{32}$$

Equation equation 32 is a decomposition of the objective, not a replacement for the implementation's exact loss specification.

In particular, once the auxiliary term is included, the minimizer of the complete objective need not equal equation 26. The conditional expectation result characterizes the quadratic component only.

A.1.3 Connection to pixel-space clean prediction

Li & He (2026) investigate the distinction between predicting clean images and directly predicting noisy quantities, including Gaussian noise and flow velocity. Their central motivation is the manifold hypothesis: natural images may occupy a comparatively low-dimensional subset of their high-dimensional pixel space, whereas Gaussian noise and quantities containing it are not similarly restricted. Under this hypothesis, predicting clean data can allow a network with a representational bottleneck to discard irrelevant noisy directions rather than reconstructing them explicitly.

NucleCast adopts the same clean-prediction principle but applies it to a conditional spatiotemporal residual. The frozen deterministic model produces the unclipped forecast $\widehat{Y}_{\text{anchor}}$, from which the dataset-specific clipped anchor D and conditioning mask M are constructed as defined above. The flow model does not predict the complete future field directly; it predicts the clean residual

$$R = Y - D. \tag{33}$$

Accordingly, at an intermediate flow state X_t the network output is

$$\widehat{R}_t = f_\theta(X_t, t, C), \qquad C = (x_{\text{past}}, D, M), \tag{34}$$

and the full forecast is reconstructed only after integrating the residual transport and adding the terminal residual state back to the anchor.

A.1.4 Conditional forecasting and temporal dimensionality

A radar forecast is not simply an unconditional image with additional channels.

For any context C, the target distribution is

$$p(Y \mid C), \tag{35}$$

and the flow must represent possible future evolutions compatible with the same observed history.

For a sequence of T future frames, each with $H \times W$ spatial resolution,

$$Y \in \mathbb{R}^{THW} \tag{36}$$

for a single radar variable.

The conditional covariance is a spatiotemporal block matrix,

$$\Sigma_Y(c) = \text{Cov}(Y \mid C = c), \tag{37}$$

whose off-diagonal blocks describe correlations between forecast lead times.

The future distribution may contain uncertainty about displacement, intensification, decay, splitting, and the appearance of convective structures. Different realizations can consequently differ in location and timing as well as amplitude. These properties distinguish the forecasting task from unconditional natural-image generation.

A.1.5 RESIDUALIZATION IS A CONDITIONAL TRANSLATION

The deterministic anchor has a precise mathematical effect on the target distribution.

Fix $C = c$ and write

$$R = Y - D(c). \tag{38}$$

Since $D(c)$ is constant after conditioning on $C = c$, residualization is a translation of the conditional distribution:

$$p_R(A \mid c) = p_Y(A + D(c) \mid c) \tag{39}$$

for every measurable set A.

In particular,

$$\mathbb{E}[R \mid C = c] = \mathbb{E}[Y \mid C = c] - D(c). \tag{40}$$

The conditional covariance is unchanged:

$$\boxed{\mathrm{Cov}(R \mid C = c) = \mathrm{Cov}(Y \mid C = c).} \tag{41}$$

Indeed, defining $\mu_Y(c) = \mathbb{E}[Y \mid C = c]$ gives

$$\begin{aligned} R - \mathbb{E}[R \mid C = c] &= Y - D(c) - \mu_Y(c) + D(c) \\ &= Y - \mu_Y(c), \end{aligned} \tag{42}$$

which proves equation 41.

Translation also preserves pairwise distances between realizations:

$$\|R_1 - R_2\| = \|Y_1 - Y_2\| \tag{43}$$

for two possible futures under the same context.

Thus residualization does not automatically reduce conditional uncertainty, remove conditional modes, or reduce the intrinsic dimension of the conditional target distribution.

It can nevertheless reduce the unconditional magnitude of the learning target.

For fixed c,

$$\boxed{\mathbb{E}[\|R\|^2 \mid C = c] = \mathrm{tr}\,\Sigma_Y(c) + \|\mu_Y(c) - D(c)\|^2.} \tag{44}$$

The conditional-mean anchor

$$D(c) = \mu_Y(c) \tag{45}$$

minimizes this expected squared residual magnitude among deterministic anchors, but leaves the conditional covariance unchanged.

The unconditional variance decomposition makes the distinction explicit:

$$\mathrm{Cov}(R) = \mathbb{E}[\mathrm{Cov}(Y \mid C)] + \mathrm{Cov}(\mu_Y(C) - D(C)). \tag{46}$$

An accurate deterministic stage can reduce the second term, which describes systematic variation in residual means across contexts. It does not remove the first term, which describes uncertainty remaining after the context is known.

A.1.6 THE EFFECT OF ANCHOR CLIPPING.

The implemented residual is defined relative to a clipped anchor.

Let $\widehat{D}(C)$ denote the unclipped deterministic forecast, expressed in the same numerical coordinates as Y and $D(C)$. Then

$$\boxed{R = \underbrace{Y - \widehat{D}(C)}_{\text{forecast error}} + \underbrace{\widehat{D}(C) - D(C)}_{\text{clipped-off anchor component}}.} \tag{47}$$

Consequently, even if the unclipped predictor approximates $\mu_Y(C)$, the mean residual need not vanish where clipping removes intensity:

$$\mathbb{E}[R \mid C] \approx \widehat{D}(C) - D(C). \tag{48}$$

The approximation $\mathbb{E}[R \mid C] \approx 0$ is justified only where the actual residual anchor $D(C)$ approximates the conditional mean.

It must not be asserted globally merely because the unclipped forecaster is accurate.

A.1.7 Architectural interpretation and reconstruction

Residualization is not intended to remove the intrinsic stochasticity of forecasting. Its architectural role is to assign different computational tasks to the two stages.

The frozen deterministic model produces

$$D(C), \tag{49}$$

which supplies the large-scale forecast and a reference for the residual.

The generative stage models

$$p_R(R \mid C) \tag{50}$$

and samples corrections without having to produce the entire forecast from an unstructured representation.

This is a division of modeling responsibility, not a theorem that residual generation is always easier or more accurate.

There is also a useful exact equivalence in the idealized unprocessed coordinates. Adding the deterministic anchor to equation 18 gives

$$\begin{aligned} D(C) + X_t &= D(C) + (1-t)Z + t(Y - D(C)) \\ &= (1-t)(D(C) + Z) + tY. \end{aligned} \tag{51}$$

Thus the residual path $Z \to Y - D(C)$ is the full-field path

$$D(C) + Z \to Y \tag{52}$$

expressed in translated coordinates.

Simply replacing residual prediction by full-field prediction while translating the source by the same anchor does not define a different straight transport problem.

In the actual reconstruction pipeline, clipping and standardization must be applied in the order specified by the implementation. If $\mathcal{T}$ denotes the preprocessing map, the model-space residual is formed using the consistently preprocessed target and anchor. Reconstruction adds the model-space anchor to the generated residual before applying the corresponding inverse transformation.

The translation identities above apply exactly in the common model coordinates. Any subsequent nonlinear clipping, physical range enforcement, quantile mapping, or ensemble reduction constitutes an additional operation and need not preserve the distributional identities of the ideal flow.

A.1.8 Scope of the identities

The derivations in this appendix require finite second moments for the squared-error statements and use the exact straight interpolation in equation 18.

The velocity conversion equation 23 is defined for $t < 1$. At $t = 1$, its denominator vanishes; using an endpoint value requires an appropriate limiting or separately defined field.The conditional-expectation minimizer is defined almost everywhere under the corresponding intermediate-state distribution. If the training-time distribution has a continuous density and never samples $t = 0$ exactly, the endpoint value at $t = 0$ is an idealized fixed-time result, not a guarantee about the trained network's output at an unobserved time.

The full training objective, finite model capacity, stochastic optimization, auxiliary wavelet loss, discretized ODE solver, and reconstruction pipeline can all produce deviations from the population-optimal identities.

Accordingly, clean prediction should be understood here as a modeling and parameterization choice supported by the natural-image results of Li and He, not as a guarantee of accurate conditional radar generation.

A.2 Wavelet-Domain Detail Penalty

Standard squared-error velocity matching targets the conditional expectation of the velocity field, which naturally induces spatial smoothing on high-frequency structures. To preserve sharp convective gradients without destabilizing the flow trajectory, we augment the objective with an auxiliary wavelet-domain penalty $\mathcal{L}_{\mathrm{wav}}$ on the clean residual prediction $\widehat{R}_t$.

Motivation and Parseval Equivalence. Under an orthonormal wavelet basis with uniform L_2 weights across subbands, Parseval's identity implies exact equivalence to pixel-space MSE:

$$\sum_{j,k} |\langle x, \psi_{j,k}\rangle|^2 = \|x\|_2^2. \tag{53}$$

An effective frequency-selective regularizer therefore requires breaking this isometric equivalence (Korkmaz et al., 2024). We achieve this by: (i) adopting an L_1 norm rather than L_2 to encourage sparsity in high-frequency detail coefficients and avoid penalizing isolated convective edges quadratically, and (ii) enforcing non-uniform, scale-dependent subband weights that emphasize fine turbulent textures.

Mathematical Formulation. Formally, $\mathcal{L}_{\mathrm{wav}}$ is defined as:

$$\mathcal{L}_{\mathrm{wav}}(\widehat{R}_t, R) = w_{\mathrm{LL}} \left\| A_J(\widehat{R}_t) - A_J(R) \right\|_1 + \sum_{\ell=1}^{J} w_\ell \left\| D_\ell(\widehat{R}_t) - D_\ell(R) \right\|_1, \tag{54}$$

where $\big(A_J(\cdot), D_1(\cdot), \ldots, D_J(\cdot)\big) = \mathcal{W}_J(\cdot)$ denotes a J-level decimated 2D Discrete Wavelet Transform using Daubechies-2 wavelets (Daubechies, 1988) with $J = 3$. The transform is evaluated per frame across the batch and lead dimensions.

Here, A_J represents the coarsest approximation subband, while $D_\ell = (\mathrm{LH}_\ell, \mathrm{HL}_\ell, \mathrm{HH}_\ell)$ aggregates the horizontal, vertical, and diagonal detail subbands at scale ℓ. The subband norm $\|\cdot\|_1$ denotes the mean absolute error across all coefficients within that subband. Subband weights are set to down-weight large-scale structures already captured by the base velocity loss while heavily penalizing errors at fine spatial scales:

$$w_{\mathrm{LL}} = 0.1, \qquad (w_1, w_2, w_3) = (1.0, 1.5, 2.0), \tag{55}$$

indexed from coarsest ($\ell = 1$) to finest ($\ell = 3$).

Implementation Details. Both the wavelet decomposition and loss evaluation are computed in 32-bit floating point precision outside automatic mixed precision. Unlike the base velocity loss, which carries an implicit $1/(1-t)^2$ time-dependent factor, $\mathcal{L}_{\mathrm{wav}}$ is applied uniformly across flow time t. This ensures a steady gradient signal at early trajectory times ($t \to 0$), where conditional mean collapse is most severe. The loss weight is set to $\lambda_{\mathrm{wav}} = 0.8$ based on preliminary development runs.

A.3 Ablation: Performance vs. Number of Function Evaluations

A.3.1 Numerical integration and the distinction between support and selectivity

The registered NucleCast sampler uses a five-point Euler time grid and four network evaluations per ensemble member.

For grid points $t_0, \ldots, t_4$, its updates have the form

$$x_{k+1} = x_k + (t_{k+1} - t_k) v_\theta(x_k, t_k, C), \qquad k = 0, \ldots, 3. \tag{56}$$

The resulting numerical map need not reproduce the terminal distribution of the learned continuous-time field exactly. A historical solver diagnostic held the checkpoint, evaluation windows, and stochastic draws fixed while changing the numerical integrator.The following Table 4 reports that diagnostic. All rows use the same 288 development windows, one seed root and one draw per window. The escape-site denominator is the 5,285 ground-truth sites in this particular evaluation population.

Integrator	Off-support mass share	Exact hits	Radius-2 hits	CSI-M	FAR_{219}
Euler, 4 evaluations	0.0011	0	6	0.459	0.687
Euler, 19 evaluations	0.0217	1	63	0.438	0.745
Euler, 79 evaluations	0.0463	11	142	0.424	0.778
Adaptive Dopri5	0.0738	29	224	0.402	0.820

Table 4: Historical development-population solver diagnostic. The off-support mass share is the pooled fraction of predicted extreme-core pixels outside the anchor mask. Exact and radius-two columns count overlaps with observed escape sites. The table describes one-draw contrasts between solver settings, not training-seed uncertainty or a new confirmatory cohort. Ground truth off-support core mass is 0.1104.

Finer integration thus generated substantially more off-support precipitation without producing a comparable improvement in overlap with the realized escape sites. Under adaptive integration, exact recall remained below 1%, while the false-alarm ratio at the extreme threshold increased and CSI-M decreased relative to the registered sampler.

These observations establish that the apparent confinement at the registered operating point is not a hard architectural constraint. They also demonstrate that greater generated off-support mass does not necessarily yield greater spatial selectivity or better single-draw categorical skill.

Qualitative comparisons across multiple development windows (Figures 5, 6, and 7) corroborate these quantitative findings. As the number of function evaluations increases, the model progressively exhibits severe over-forecasting and generates spurious, off-support extreme precipitation patches (visible as isolated high-intensity cores outside the physical anchor mask). In contrast, the registered 4-step Euler sampler maintains a significantly cleaner and tighter spatial confinement.

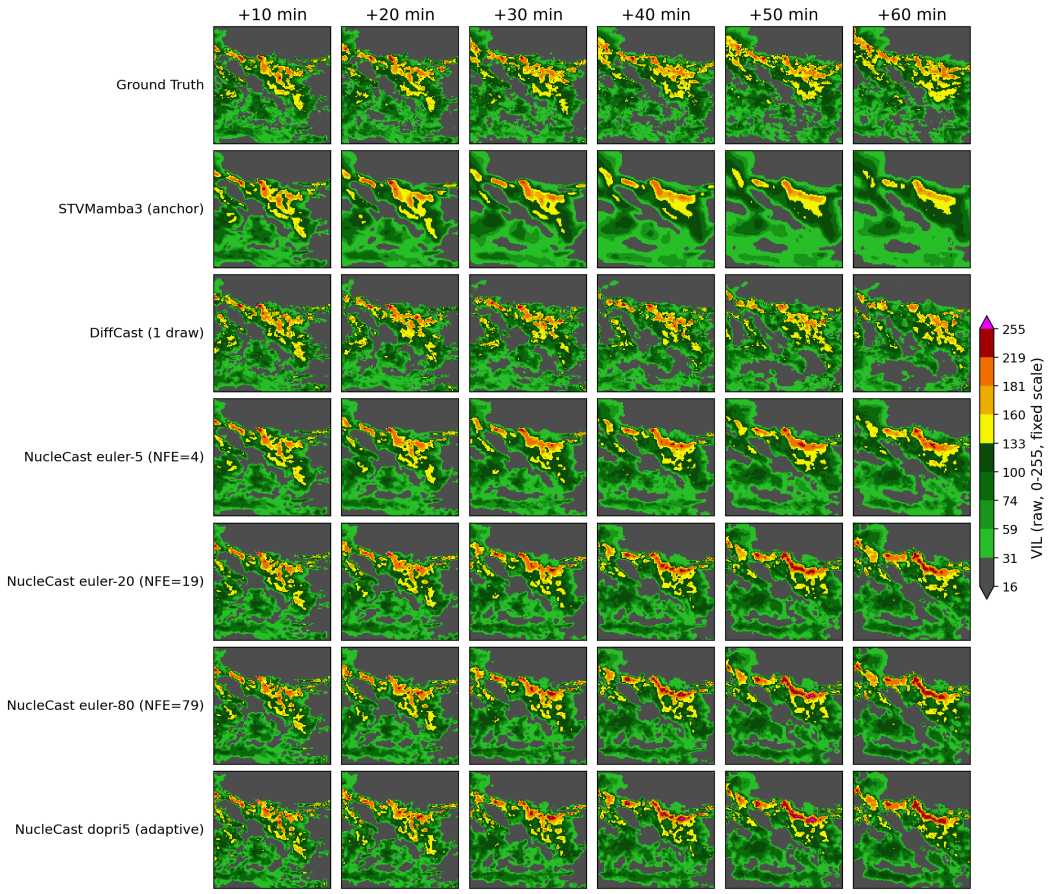


Figure 5: Qualitative NucleCast NFE ablation comparison on SEVIR. (window 90)

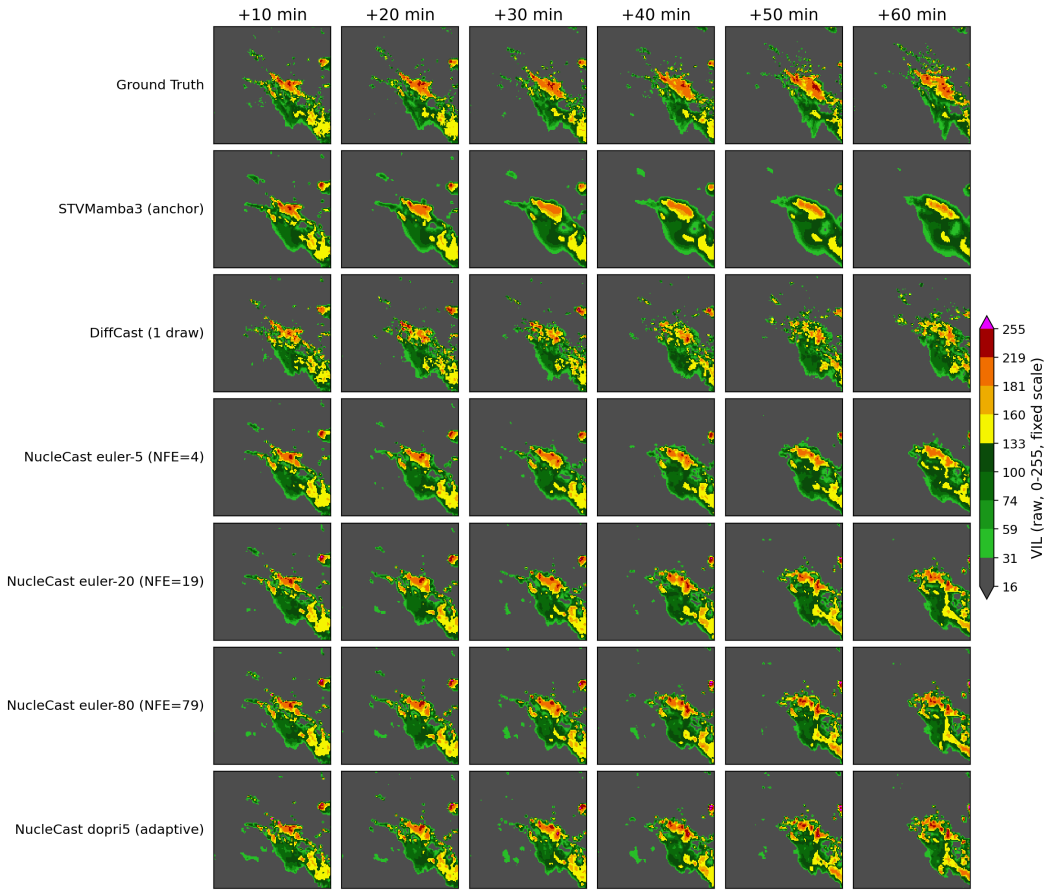


Figure 6: Qualitative NucleCast NFE ablation comparison on SEVIR. (window 1362)

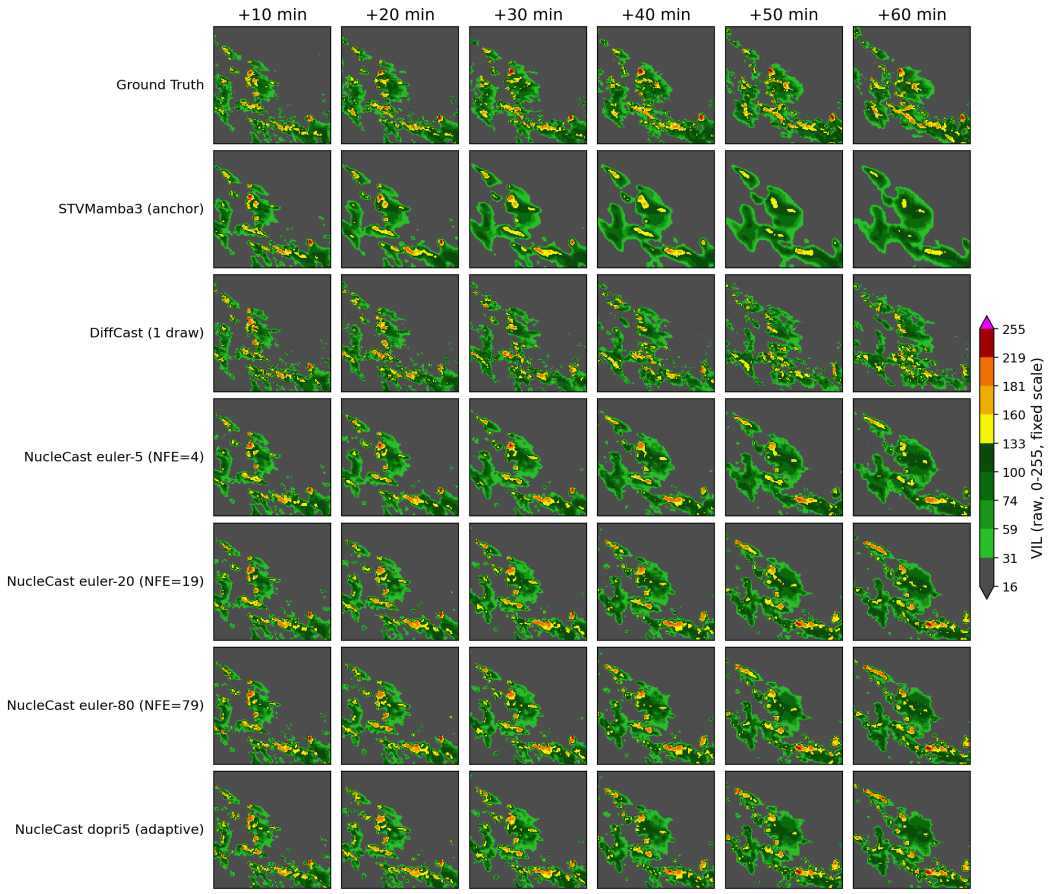


Figure 7: Qualitative NucleCast NFE ablation comparison on SEVIR. (window 2436)

What the solver experiment identifies. Because the checkpoint and stochastic inputs were held fixed, the experiment isolates the effect of changing the numerical integration procedure on the resulting samples. The comparison across Euler evaluation counts provides evidence of step-size sensitivity. Adaptive Dopri5 additionally changes the integration algorithm.

The experiment does not independently establish the exact continuous-time terminal law, prove numerical convergence, or identify which component of the learned-field error is responsible for incorrect localization. In particular, observing more off-support precipitation with finer integration does not prove that the source law or coupling is correct. Conversely, poor spatial overlap of individual generated samples with a single observed future does not, by itself, establish that every generated alternative is implausible under the true conditional distribution. A full probabilistic assessment additionally requires ensemble-based calibration and proper distributional verification.

A.4 Deterministic Stage-1 Model: Architectural Ablation

A.4.1 Purpose

In a two-stage predictive nowcasting framework, the stage-1 deterministic model serves as an anchoring backbone that generates temporally coherent, baseline trajectory forecasts from preceding observational frames, providing an informative prior that conditions subsequent stochastic diffusion or flow-matching refinement. To determine the most effective architecture for this deterministic stage, we conduct a systematic ablation across five candidate models: four variants of the state-space architecture with Spatio-Temporal Selective Scans (STSS) and the Spatio-Temporal Dense Streams (STDS) with different numbers of spatio-temporal directions, namely $\mathrm{STVMamba3}_{\mathrm{STDS+STSS}_6}$, $\mathrm{STVMamba3}_{\mathrm{STDS+STSS}_4}$, $\mathrm{STVMamba3}_{\mathrm{STSS}_6}$, and $\mathrm{STVMamba3}_{\mathrm{STSS}_4}$, alongside the estab-

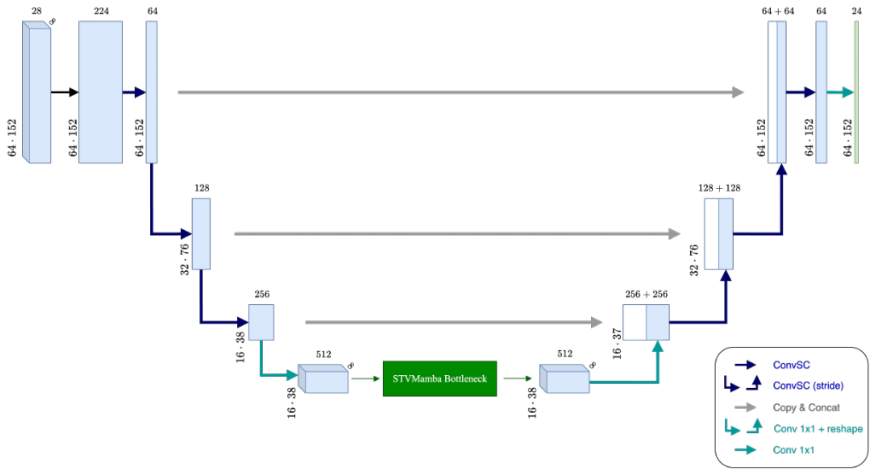


Figure 8: Architecture of the STVMamba3 block, depicting the dual-branch formulation comprising the Spatio-Temporal Selective Scan (STSS) and the parallel Spatio-Temporal Dense Stream (STDS).

lished convolutional attention baseline $\mathrm{SimVP}_{\mathrm{gSTA}}$. This comparative study investigates three core architectural questions: first, whether introducing a parallel STDS stream provides complementary inductive biases that improve upon selective state-space scans alone; second, whether expanding the scan trajectory from 4 to 6 spatio-temporal directions captures richer cross-axis dependencies; and third, whether selective state-space models outperform convolutional spatial-temporal attention networks in predictive accuracy and parameter efficiency.

A.4.2 Model Architectures

The proposed STVMamba3 backbone integrates selective state-space duality into a modular encoder-translator-decoder paradigm, as illustrated in Figure 8. The network processes input frame sequences by first mapping spatial fields into a latent representation through convolutional stem stages. Within the translator, the core computational engine alternates between two mechanisms: the STSS and the STDS. The STSS mechanism flattens spatio-temporal feature volumes into sequence orderings and applies data-dependent selective state-space transformations with linear computational complexity in sequence length. In the 4-direction configuration (STSS_4), scans traverse spatial dimensions along standard forward and reverse raster paths. In the 6-direction configuration (STSS_6), directional coverage is augmented with explicit forward and backward temporal sweeps across frames, ensuring coupled past-future and multi-axis context exchange. The complementary STDS stream introduces a parallel 3D depthwise convolutional pathway that directly extracts localized spatio-temporal correlations, preventing high-frequency feature dissipation across deep layers. In contrast, the baseline $\mathrm{SimVP}_{\mathrm{gSTA}}$ model utilizes a pure convolutional translator governed by spatial-temporal attention (gSTA), which captures frame interactions through large-kernel depthwise convolutions and gating units without recurrent hidden state vectors.

A.4.3 Data and Training Protocol

To evaluate intrinsic spatio-temporal modeling capabilities in isolation from domain-specific noise, the ablation study was conducted on the MovingMNIST benchmark (rendered on-the-fly with 2 moving digits per sequence at 64×64 grayscale resolution) (Srivastava & Salakhutdinov, 2015), rather than on real-world meteorological radar data such as SEVIR VIL. This synthetic setting provides an ideal testbed because it presents complex non-linear dynamics, occlusions, and multi-body interactions under strictly controlled kinematic rules, free from confounding precipitation artifacts, topographic variability, or sensor calibration noise. Models are trained to predict $T_{\mathrm{out}} = 10$ future frames conditioned on $T_{\mathrm{in}} = 10$ past observations. All candidates are optimized under identical training conditions, batch sizes, hardware configurations, and loss functions, as detailed in Table 5.

Table 5: Training conditions and hyperparameter summary for deterministic stage-1 video-prediction models, structured in accordance with FlowCast's architecture specification. All models are trained under identical data, loss, batching, and hardware constraints.

Category / Parameter	STVMamba3 Family	$\mathbf{SimVP}_{\mathrm{gSTA}}$
Dataset		
Source	MovingMNIST (rendered on-the-fly)	MovingMNIST (rendered on-the-fly)
Digits	2 moving digits (28×28 glyphs)	2 moving digits (28×28 glyphs)
Input Frames (T_{in})	10	10
Output Frames (T_{out})	10 (also evaluated at 50)	10 (also evaluated at 50)
Spatial Resolution	64×64, 1 channel (grayscale)	64×64, 1 channel (grayscale)
Dataset Size	10,000 train / 1,000 val / 1,000 test	10,000 train / 1,000 val / 1,000 test
Training Configuration		
Loss Function	Mean Squared Error ($\mathcal{L}_{\mathrm{MSE}} = \|\hat{y} - y\|^2$)	Mean Squared Error ($\mathcal{L}_{\mathrm{MSE}} = \|\hat{y} - y\|^2$)
Max. Epochs	200 (converged) / 50 (pre-ablation scan)	200
Batch Size	16 (Global = 16, Local = 16)	16 (Global = 16, Local = 16)
Precision	`bf16-mixed`	`bf16-mixed`
Gradient Clipping	Global Norm 1.0	Global Norm 1.0
Hardware	$1 \times$ NVIDIA GPU (PCIe)	$1 \times$ NVIDIA GPU (PCIe)
Wall Time	7.45 h (200 ep) / 1.27–1.76 h (50 ep)	2.21 h (200 ep)
Optimization		
Optimizer	AdamW	AdamW
Initial Learning Rate	1.0×10^{-3}	1.0×10^{-3}
Weight Decay	1.0×10^{-5}	1.0×10^{-4}
AdamW Betas	$(0.9, 0.999)$	$(0.9, 0.999)$
AdamW Epsilon	1.0×10^{-8}	1.0×10^{-8}
LR Scheduler	OneCycleLR (pct=0.3, div=25, final_div=10^4)	OneCycleLR (pct=0.3, div=25, final_div=10^4)
Architecture Parameters		
Spatial Encoder/Decoder	Conv2d ($H_s = 64, N_s = 3$)	Conv2d ($H_s = 64, N_s = 4$)
Translator Mechanism	Spatio-Temporal Mamba3 (STSS)	Spatial-Temporal Attention (gSTA)
Hidden Dimension	$C_1 = 64, C_2 = 96$	$H_t = 512, N_t = 8$
State Dimension (d_{state})	128	N/A
Scan Directions	4 or 6 spatio-temporal directions	N/A
STDS Conv Stream	Depthwise 3D Conv ($k = 3$) / Disabled	N/A
Parameter Count	9.40 M (STDS) / 9.38 M (noSTDS)	46.80 M

A.4.4 Validation Metrics

Model performance is evaluated across four validation metrics:

- **Mean Squared Error (MSE)**: Measures the average squared Euclidean difference between predicted and ground-truth pixel values, heavily penalizing large deviations.
- **Mean Absolute Error (MAE)**: Calculates the average absolute pixel deviation, quantifying trajectory fidelity with uniform sensitivity to errors.
- **Structural Similarity Index (SSIM)**: Evaluates perceptual image quality by assessing local luminance, contrast, and structural correlation across sliding windows.
- **Peak Signal-to-Noise Ratio (PSNR)**: Computes the logarithmic ratio between peak signal energy and mean squared reconstruction error, expressed in decibels (dB).

The empirical validation results are summarized in Table 6, and the corresponding validation convergence trajectories across training epochs are depicted in Figure 9.

A.4.5 Conclusion

The quantitative evaluations in Table 6 establish a clear performance ranking across the candidate architectures. In the matched 50-epoch factorial comparison, $\mathrm{STVMamba3}_{\mathrm{STDS+STSS}_6}$ achieves the best performance across all metrics with a validation MSE of 0.01619, MAE of 0.04097, and SSIM of 0.7870, followed by $\mathrm{STVMamba3}_{\mathrm{STDS+STSS}_4}$ (MSE 0.01748, MAE 0.04389, SSIM 0.7629), $\mathrm{STVMamba3}_{\mathrm{STSS}_6}$ (MSE 0.01779, MAE 0.04463, SSIM 0.7671), $\mathrm{SimVP}_{\mathrm{gSTA}}$ (MSE 0.01847, MAE 0.04696, SSIM 0.6924, PSNR 17.77), and $\mathrm{STVMamba3}_{\mathrm{STSS}_4}$ (MSE 0.02289, MAE 0.05631, SSIM 0.6320). Upon full convergence at 200 epochs, $\mathrm{STVMamba3}_{\mathrm{STDS+STSS}_6}$ attains a validation MSE of 0.00937, MAE of 0.02636, SSIM of 0.8982, and PSNR of 20.91 dB, outperforming both the 200-epoch $\mathrm{SimVP}_{\mathrm{gSTA}}$ baseline (MSE 0.01225, MAE 0.03257, SSIM 0.8548,

Table 6: Validation performance across the five evaluated stage-1 deterministic architectures on MovingMNIST ($10 \rightarrow 10$). Metrics are reported at the final training epoch (epoch 50 for the 4-direction screening variants, epoch 200 for converged models), following the evaluation protocol established in the results documentation; within 10^{-5} MSE, this rule coincides with the minimum validation loss checkpoint. Best value per metric column is shown in **bold**. The lower block presents the strictly matched 50-epoch checkpoint across all five models.

Model	Params (M)	Epochs	SE ↓	MAE ↓	SSIM ↑	PSNR (dB) ↑
Terminal / Converged Benchmark						
$\text{STVMamba3}_{\text{STDS+STSS}_6}$	9.40	200	**0.00937**	**0.02636**	**0.8982**	**20.91**
$\text{STVMamba3}_{\text{STDS+STSS}_4}$	9.40	50	0.01748	0.04389	0.7629	—
$\text{STVMamba3}_{\text{STSS}_6}$	**9.38**	200	0.01311	0.03458	0.8413	—
$\text{STVMamba3}_{\text{STSS}_4}$	**9.38**	50	0.02289	0.05631	0.6320	—
$\text{SimVP}_{\text{gSTA}}$	46.80	200	0.01225	0.03257	0.8548	19.64
Matched 50-Epoch Factorial Pre-Ablation Checkpoint						
$\text{STVMamba3}_{\text{STDS+STSS}_6}$	9.40	50	**0.01619**	**0.04097**	**0.7870**	—
$\text{STVMamba3}_{\text{STDS+STSS}_4}$	9.40	50	0.01748	0.04389	0.7629	—
$\text{STVMamba3}_{\text{STSS}_6}$	**9.38**	50	0.01779	0.04463	0.7671	—
$\text{STVMamba3}_{\text{STSS}_4}$	**9.38**	50	0.02289	0.05631	0.6320	—
$\text{SimVP}_{\text{gSTA}}$	46.80	50	0.01847	0.04696	0.6924	**17.77**

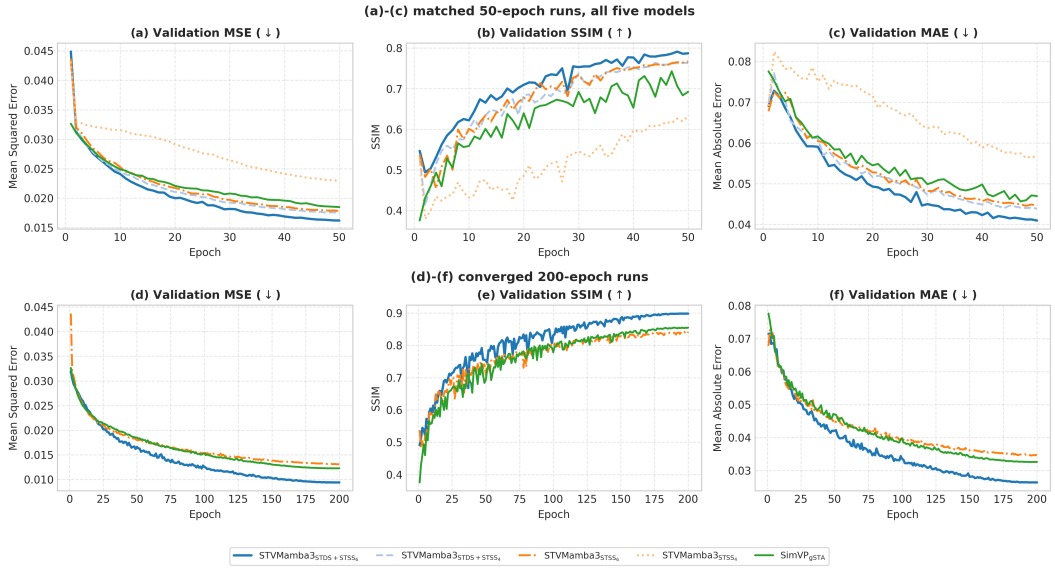


Figure 9: Validation convergence dynamics on MovingMNIST ($10 \rightarrow 10$) across stage-1 candidate models. Top row (a–c), matched 50-epoch runs across all five models: (a) Validation MSE (↓), (b) Validation SSIM (↑), and (c) Validation MAE (↓). Bottom row (d–f), converged 200-epoch runs for $\text{STVMamba3}_{\text{STDS+STSS}_6}$, $\text{STVMamba3}_{\text{STSS}_6}$, and $\text{SimVP}_{\text{gSTA}}$: (d) Validation MSE (↓), (e) Validation SSIM (↑), and (f) Validation MAE (↓).

PSNR 19.64 dB) and the scan-only $\text{STVMamba3}_{\text{STSS}_6}$ model (MSE 0.01311, MAE 0.03458, SSIM 0.8413). Notably, $\text{STVMamba3}_{\text{STDS+STSS}_6}$ achieves this superior accuracy with only 9.40 M parameters compared to 46.80 M for $\text{SimVP}_{\text{gSTA}}$, representing a near five-fold reduction in parameter budget. Based on these empirical results, $\text{STVMamba3}_{\text{STDS+STSS}_6}$ is elected as the deterministic stage-1 backbone: it consistently delivers the lowest validation error and highest structural fidelity across all training epochs while using less than a quarter of the parameters of the convolutional baseline. We note the caveat that this finding is based on validation split evaluations.

A.5 SPADE MAMBA BLOCK

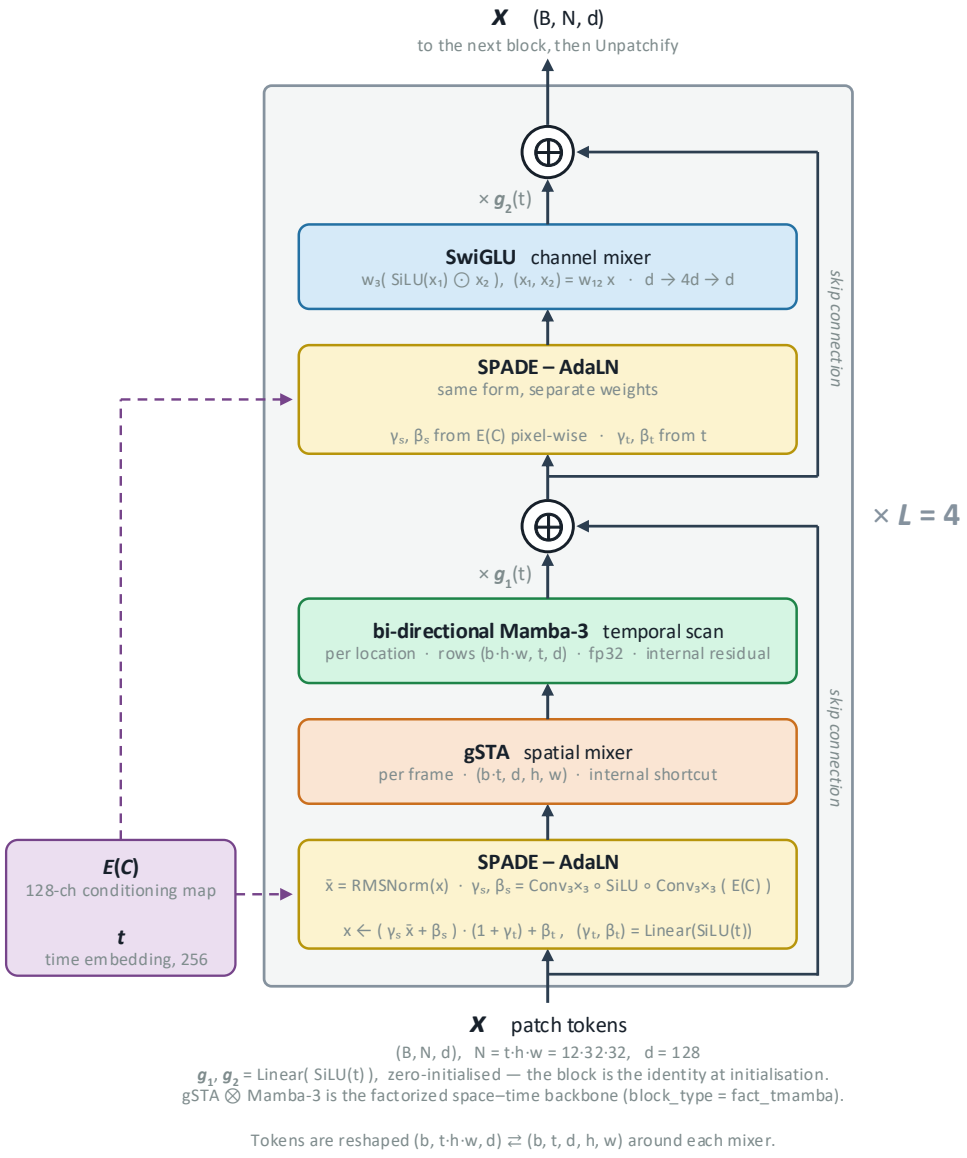


Figure 10: **Architecture of the SPADE–Mamba block.** Each block applies spatial mixing through gSTA followed by bidirectional temporal modeling with Mamba-3. SPADE–AdaLN modulates the features using the spatial conditioning map $E(C)$ and the flow-time embedding. A SwiGLU channel mixer completes the block, with gated residual connections around the temporal and channel-mixing branches. Four such blocks are stacked in the clean-residual prediction network f_θ.

A.6 NUCLECAST'S SOURCE DISTRIBUTION ABLATION

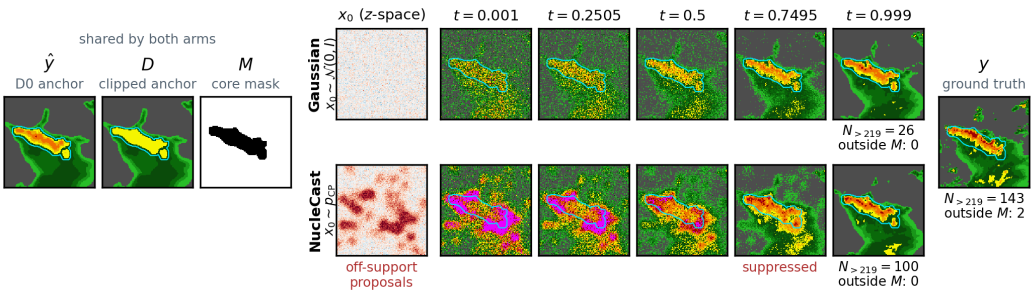


Figure 11: **NucleCast matched source ablation (window 122, lead 10/12)**: same anchor, conditionning, target sampler and t-grid. only the source law x_0 differs

A.7 CASCADE REFINEMENT

Cohort S (Storm Initiation) is a subset of the SEVIR test set (641 timestamps, 4,332 sequences), drawn via proportional month $\times$ event-type stratified sampling of the test set. Rather than evaluating pre-existing systems, Cohort S specifically isolates *de novo* storm initiation—convective cores that form outside the historical radar footprint of past observations.

Rare-Event Birth Localisation. On Cohort S, we evaluate storm initiation outside a common pre-existing convective envelope M_{common}, formed by the union of radar echoes exceeding VIL 133 across the 5 past context frames, dilated by a Chebyshev radius $r = 2$ (6 km). Observed births B_{obs} are pixels exceeding VIL 219 outside M_{common} (161,114 pixels across 35,479 storm cells). We evaluate:

- **Recall R_2:** Fraction of observed births intercepted by a predicted off-support core within 6 km ($\mathcal{D}_2$).
- **Precision $r = 2$:** Fraction of predicted off-support cores that match an observed birth within 6 km.
- **Equal-Footprint R_2:** Intercepted birth recall interpolated at a standardized predicted budget of $k = 2.0 \times |B_{\text{obs}}|$, ensuring volume-unbiased spatial localisation comparison.

Cascade Pipeline	Anchor Missed (site-pixels)	Recovery $R_2 \uparrow$	New True (tp_{new}) $\uparrow$ (components)	New False (fp_{new}) $\downarrow$ (components)	RFR $\uparrow$ ($tp_{\text{new}}/fp_{\text{new}}$)
STVMamba3 $\rightarrow$ **NucleCast (ours, 1 draw)**	146,435	0.2511	4,683	6,759	0.693
STVMamba3 $\rightarrow$ **NucleCast (ours, PM8)**	146,435	0.2489	3,976	4,285	0.928
PhyDNet $\rightarrow$ DiffCast (1 draw)	160,320	0.3196	8,369	17,642	0.474

Table 7: **Within-model refinement diagnostics on Cohort S.** Evaluated on the common B_{obs} population. Cross-cascade bolding and ranking are omitted because STVMamba3 and PhyDNet exhibit distinct anchor failure sets.

A.8 LONG HORIZON FEEDBACK FORECASTS

Autoregressive feedback modes. For multi-block rollouts (12 $\rightarrow$ 36 leads), we consider three autoregressive strategies: *output-only* (**NucleCast**$_{\text{D}}$), where refinement is applied only to reported frames while the coarse anchor rolls out on its own context; *full feedback* (**NucleCast**$_{\text{F}}$), which feeds the sharpened PM-mean back as stage-1 input; and *gated feedback* (**NucleCast**$_{\text{G}}$). In **NucleCast**$_{\text{G}}$, leads 1–5 feed back the unsharpened coarse forecast, switching to the sharpened PM-mean from lead 6 onward. This prevents early-lead stochastic noise from accumulating across autoregressive transitions while retaining sharp convective structures at longer leads.

Route	CSI-M				CSI@219			
	Leads 1–12	Leads 13–24	Leads 25–36	All 36	Leads 1–12	Leads 13–24	Leads 25–36	All 36
Single Realisation / Deterministic								
SimVP-gSTA (deterministic)	0.4623	0.2671	0.1774	0.3022	**0.1447**	0.0091	0.0000	0.0513
STVMamba3 (anchor)	**0.4755**	**0.2699**	**0.1813**	**0.3089**	0.1434	0.0000	0.0000	0.0478
DiffCast (AR 2-pass, 1 draw)	0.4099	0.2406	0.1561	0.2688	0.1207	**0.0239**	**0.0123**	**0.0523**
Eight-Member PM-Mean Operating Point								
NucleCast$_{\mathrm{G}}$ (ours, PM8)	0.4902	0.3049	**0.2153**	0.3368	0.1761	0.0199	0.0019	0.0660
FlowCast (PM8)	**0.5142**	**0.3168**	0.2128	**0.3479**	**0.2116**	**0.0226**	**0.0115**	**0.0819**

Table 8: **Long-horizon feedback, resolved into lead blocks.** 96 events (population E-2), each system rolled autoregressively to 36 leads (three hours); mean CSI within each 12-lead block. **NucleCast$_{\mathrm{G}}$** is the locked deployment: the D0+L2b cascade with the sharpener gated on (fed back into stage 1) from lead 6 onward, raw stage-1 output feeding back before that (see appendix note on the gate derivation). NucleCast$_{\mathrm{G}}$ and FlowCast are shown at the eight-member PM-mean operating point; SimVP-gSTA, STVMamba3, and DiffCast are single-draw/deterministic, with no ensemble variant run under this protocol. Best in **bold**, second-best underlined *within each of the two blocks* – single-draw and PM8 systems are never ranked against each other (see the protocol paragraph below). Paired significance against STVMamba3 and between routes (event-clustered bootstrap) is reported separately, not reproduced here.

Long-horizon protocols. STVMamba3, SimVP-gSTA, NucleCast$_{\mathrm{G}}$, and FlowCast share one protocol: each model's native 12-lead window is rolled autoregressively three times (12→12→12→12) to cover the full 36-lead, three-hour horizon, scored on the same 96-event cohort (population E-2). DiffCast's released PhyDNet checkpoint has a different native window — 5 conditioning frames in, 20 frames out, sampled with a 250-step DDIM — so it cannot be rolled the same way. We instead run it as a 2-pass chain: pass 1 conditions on the 5 ground-truth context frames and samples 20 frames; pass 2 conditions on the *last 5 frames of pass 1's own output* and samples 20 more, giving 40 raw generated frames per event, of which only the first 36 are scored, aligning DiffCast to the identical 36-lead evaluation grid used by every other row. DiffCast is run single-draw only (no PM8 operating point): at 250 DDIM steps per pass ($\times 2$ passes $= 500$ steps per event), an 8-member ensemble would cost roughly $8\times$ this arm's already-substantial compute, the same order of magnitude as the DiffCast-PM8 arm already judged too costly to certify for the single-window comparison. Because DiffCast's row is single-draw, it is ranked only against the other single-draw/deterministic systems (SimVP-gSTA, STVMamba3), never against the two PM8 rows, per this repo's standing convention against ranking across incommensurable ensemble operating points.

A.9 More Qualitative Examples

A.9.1 SEVIR

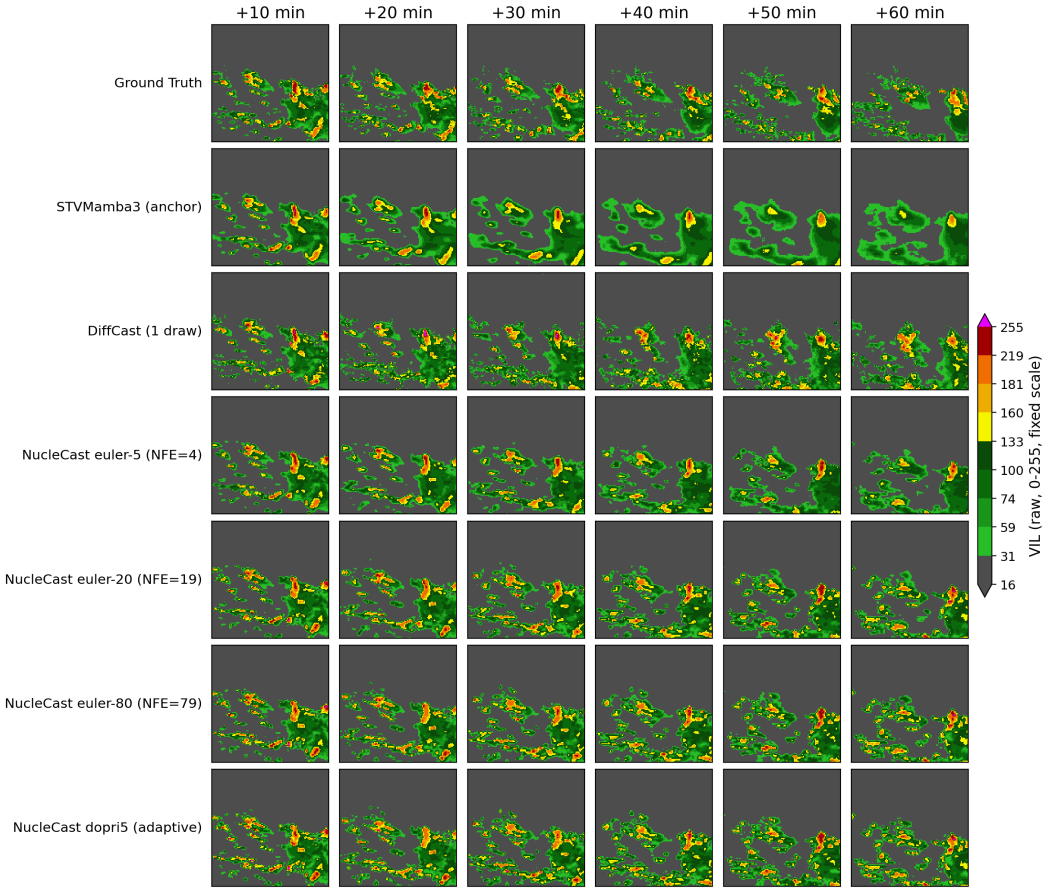


Figure 12: Qualitative comparison of NucleCast with other baselines on a SEVIR sequence.

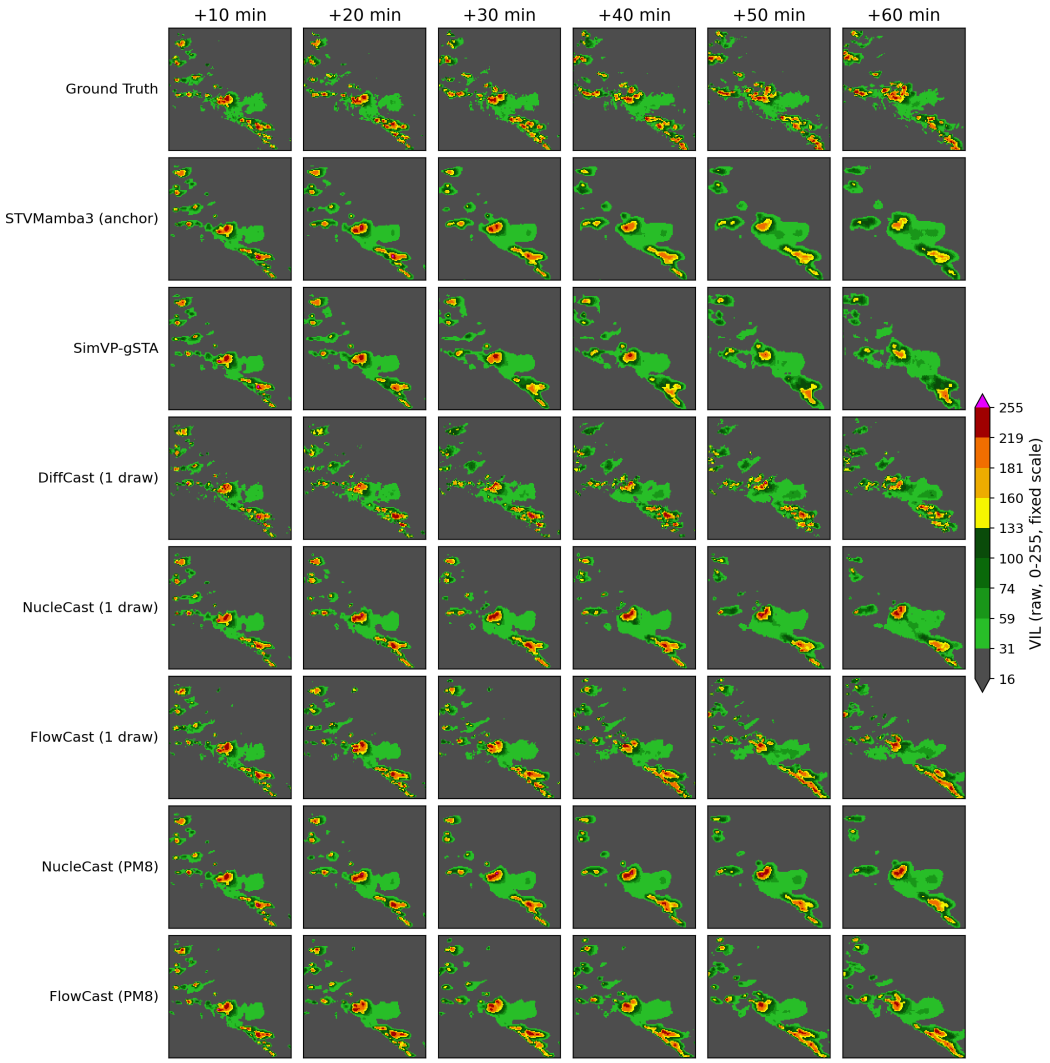


Figure 13: Qualitative comparison of NucleCast with other baselines on a SEVIR sequence.

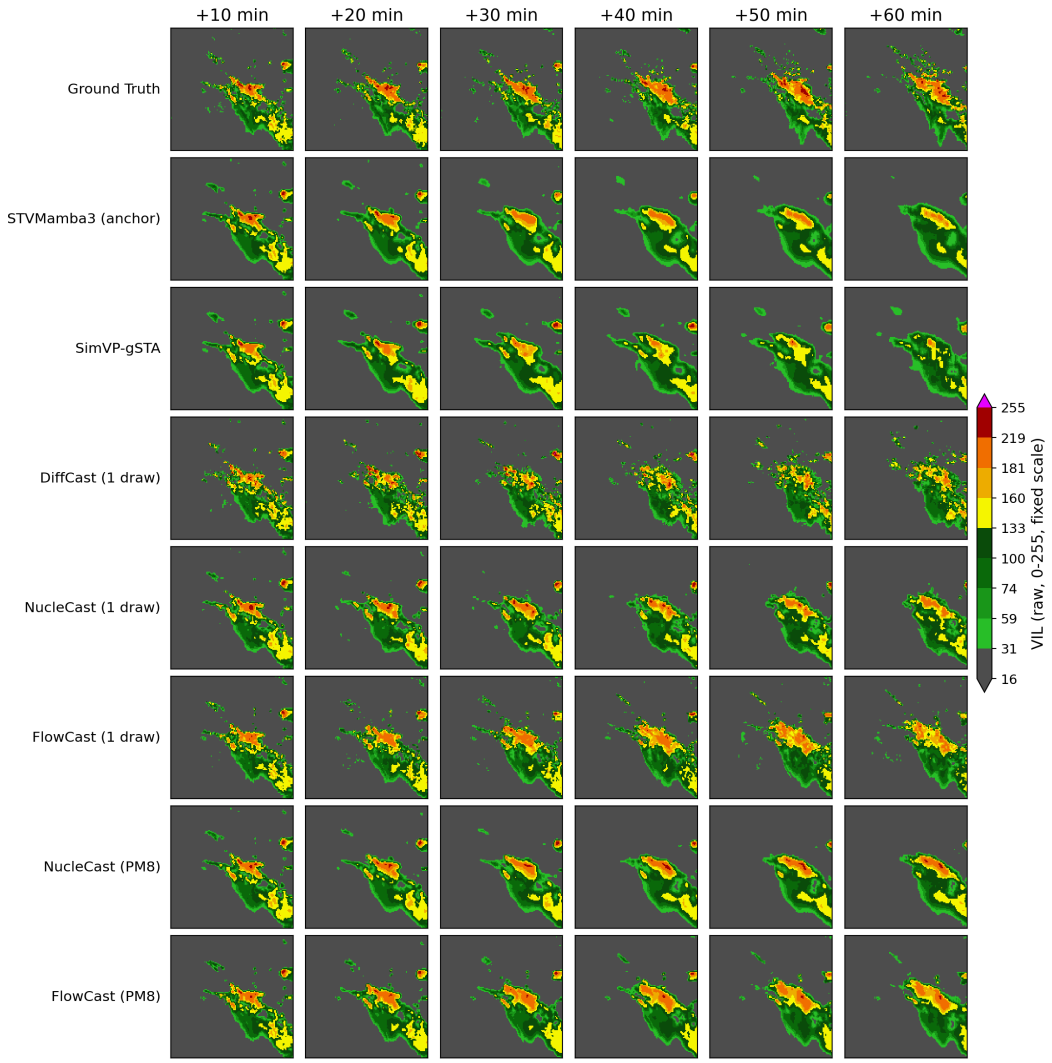


Figure 14: Qualitative comparison of NucleCast with other baselines on a SEVIR sequence.

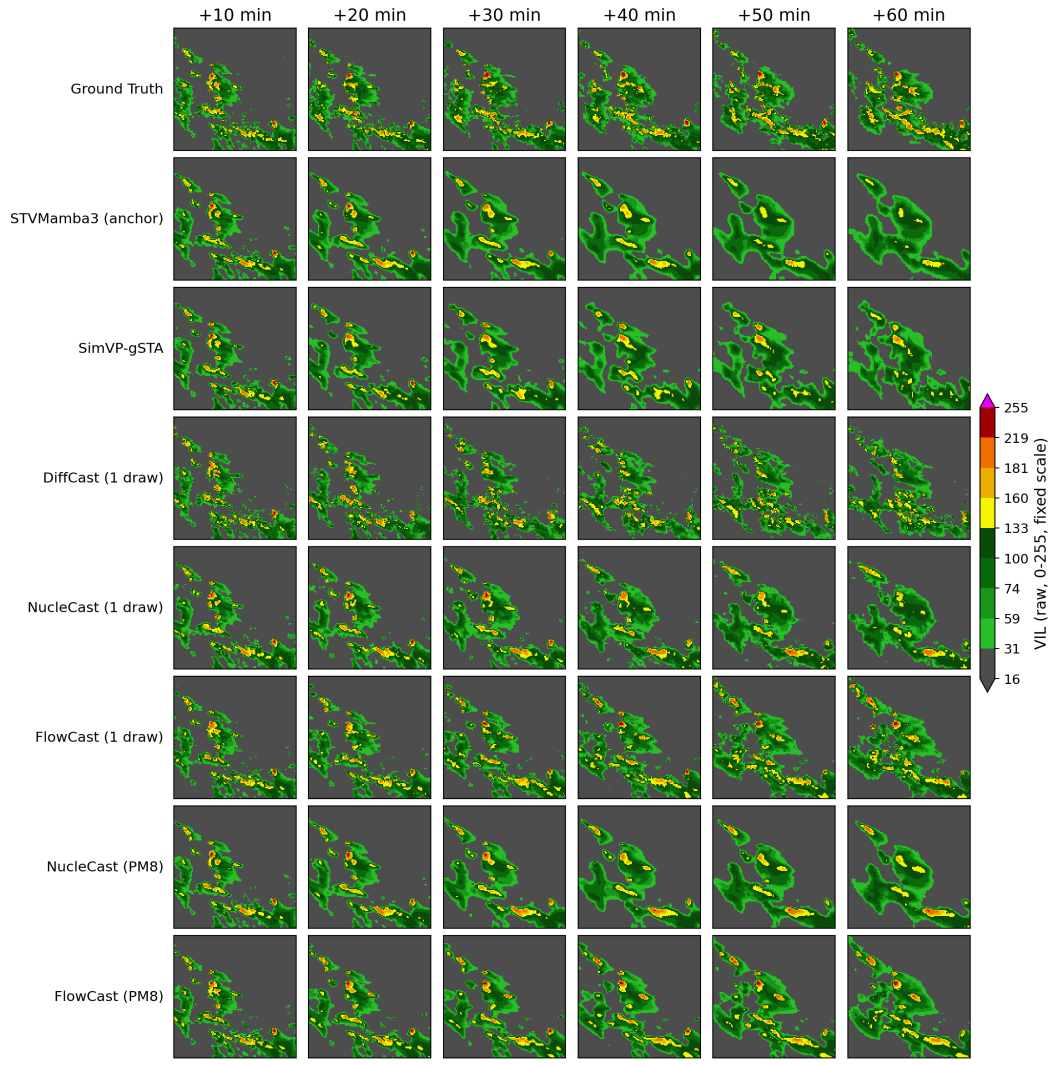


Figure 15: Qualitative comparison of NucleCast with other baselines on a SEVIR sequence.

A.9.2 ARSO

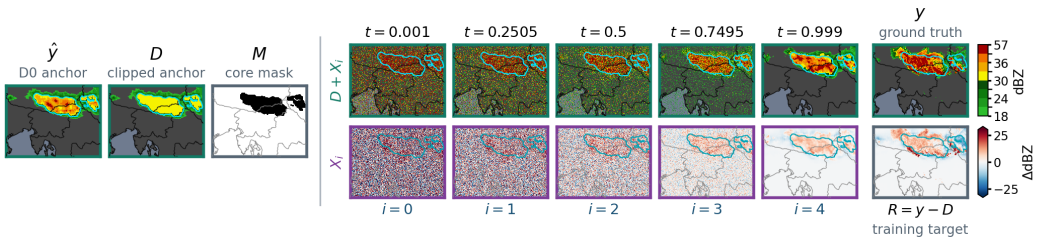


Figure 16: **NucleCast inference trajectory on ARSO (window 7946, lead 10/12). Left:** Deterministic anchor $\widehat{y}$, clipped anchor $D = D(x_{\text{past}})$ (capped at 30 dBZ), and dilated core mask M (conditioning only). **Right:** Four-step Euler rollout ($t \in [0.001, 0.999]$). *Top:* Physical reflectivity states ($D + X_i$, dBZ). *Bottom:* ODE residual states X_i, terminating near $R = y - D$. Over the rollout, the refiner removes the source noise outside the storm and builds the residual inside the core mask, restoring the convective cores above the 30 dBZ clip toward ground truth y.

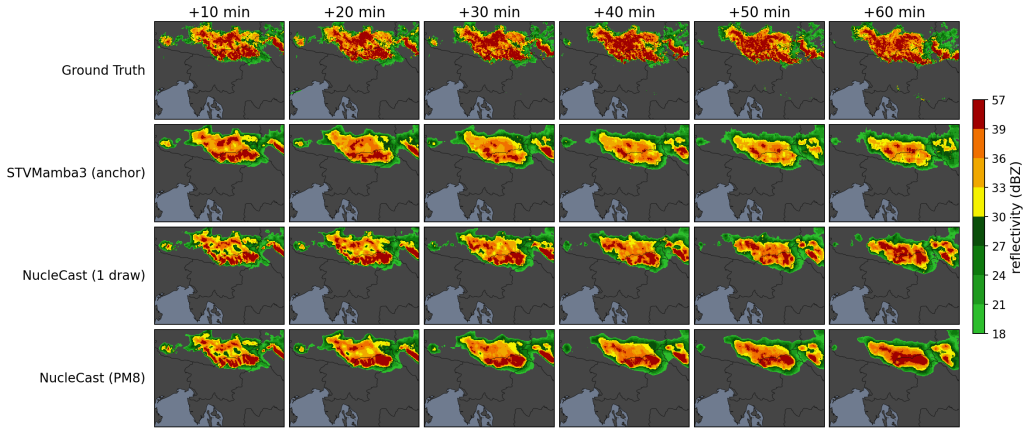


Figure 17: Qualitative comparison of NucleCast with other baselines on a ARSO sequence.

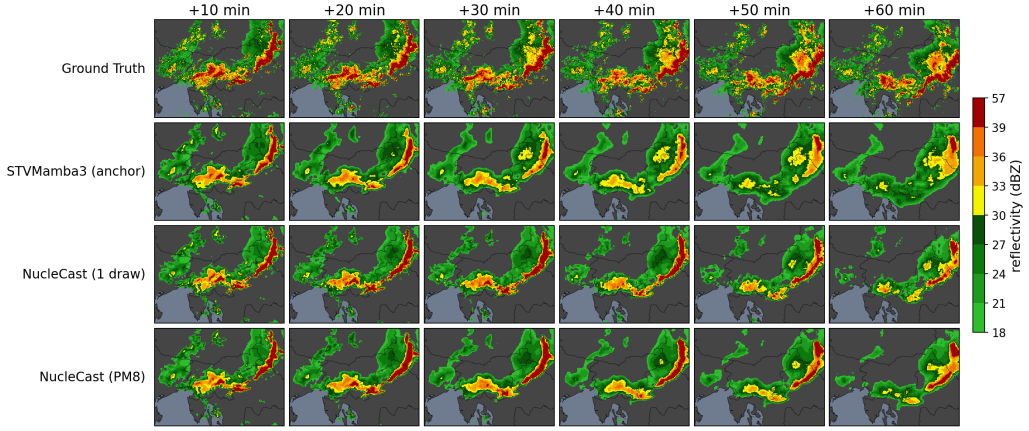


Figure 18: Qualitative comparison of NucleCast with other baselines on a ARSO sequence.

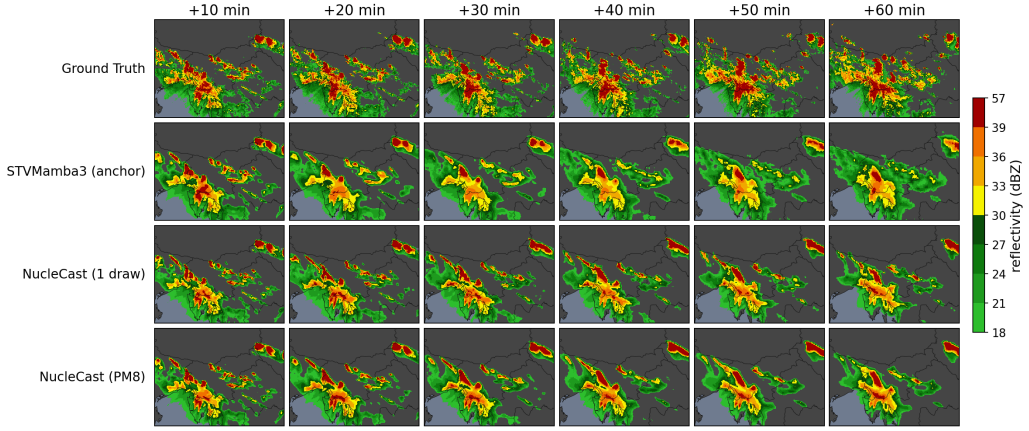


Figure 19: Qualitative comparison of NucleCast with other baselines on a ARSO sequence.

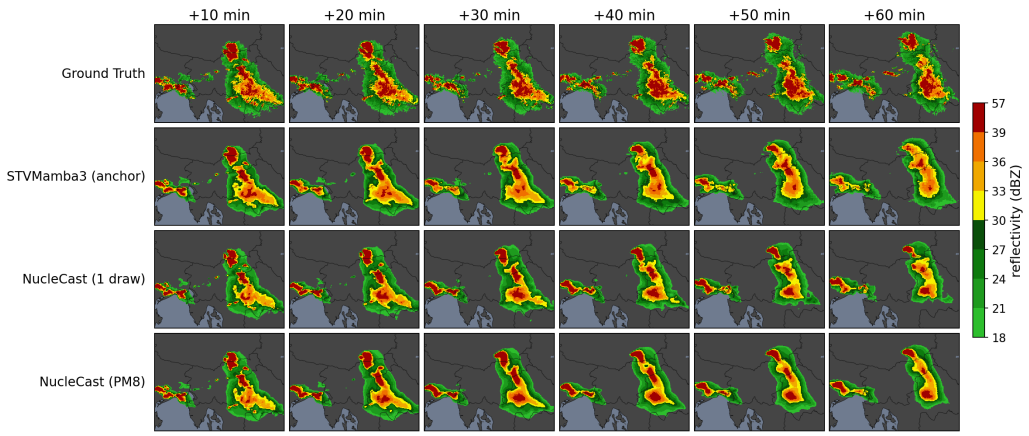


Figure 20: Qualitative comparison of NucleCast with other baselines on a ARSO sequence.