PROPER SCORING RULE-BASED DIFFUSION FOR PROBABILISTIC WEATHER FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent probabilistic weather forecasters train stochastic predictors with the continuous ranked probability score (CRPS) to generate each ensemble member in a single forward pass. These models learn the predictive distribution from the forecast context alone, which becomes difficult at longer forecast horizons where uncertainty is high. To learn the predictive distribution more effectively, we introduce auxiliary conditional denoising tasks that predict the same future state from the context and its corrupted version, which provides partial future information that can reduce prediction ambiguity. Building on distributional diffusion models, we learn the conditional distributions of these tasks with a single stochastic predictor by minimizing a proper scoring rule across noise levels. At inference, the predictor can still generate each ensemble member in a single forward pass at the fully corrupted endpoint. Standard CRPS training is recovered as the endpoint-only special case of our formulation, so our framework extends existing CRPS-based forecasters with only additional conditioning inputs. Controlled experiments show that the auxiliary tasks improve one-step forecasting across architectures, with larger gains at longer forecast horizons. The gains extend to high-dimensional global weather forecasting under both training from scratch and fine-tuning, along with improved calibration and potential benefits for generalization under distribution shift.

1 INTRODUCTION

Numerical weather prediction (NWP) has long underpinned weather forecasting by simulating atmospheric evolution. Ensembles of these simulations represent uncertainty from imperfect initial conditions and model approximations (Leutbecher & Palmer, 2008; Slingo & Palmer, 2011), but are computationally demanding at high resolution (Bauer et al., 2015). Recently, deterministic machine learning (ML) models have achieved competitive forecast skill at substantially lower inference cost than NWP (Bi et al., 2023; Lam et al., 2023), with ensemble forecasts generated from perturbed initial conditions (Kurth et al., 2023; Chen et al., 2023). However, deterministic models trained to predict the conditional mean can produce overly smooth ensemble members (Price et al., 2025). This limitation has motivated growing interest in probabilistic ML models for weather forecasting.

In probabilistic weather forecasting, models learn predictive distributions over future atmospheric states conditioned on the forecast context (Gneiting & Katzfuss, 2014). Diffusion-based forecasters generate spatially detailed ensemble members from noise through iterative conditional denoising (Price et al., 2025; Couairon et al., 2026). Yet this iterative sampling requires repeated model evaluations for each member at every forecast step, making ensemble forecasting costly. Recent CRPS-based forecasters offer an efficient alternative, training stochastic predictors with objectives based on the continuous ranked probability score (CRPS) to generate each ensemble member in a single forward pass per forecast step (Alet et al., 2025; Bonev et al., 2025; Lang et al., 2026). Such one-step ensemble generation is therefore promising for probabilistic weather forecasting.

However, these forecasters learn the predictive distribution from the forecast context alone, and such context-only training can become harder as forecast uncertainty grows. When forecasting several days ahead in one step, for example, we observe that CRPS-trained models tend to overfit (Section 3.1). To learn the predictive distribution more effectively, we draw on evidence that jointly learning related tasks can enhance performance on the target task (Caruana, 1997; Maurer et al., 2016) and consider auxiliary prediction tasks while retaining one-step forecasting. This leads to the following question.

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

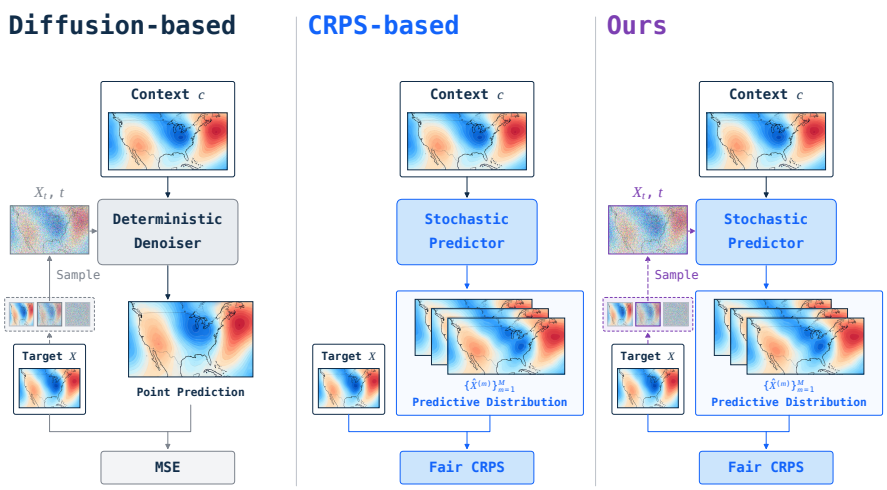


Figure 1: **Comparison of probabilistic weather forecasting methods**. Diffusion-based models (left) train a deterministic denoiser, while CRPS-based models (middle) train a stochastic predictor to generate ensembles. Our method (right) retains the stochastic predictor and adds a sampled corrupted future and its noise level as conditioning inputs (***purple***). These inputs introduce auxiliary conditional denoising tasks learned through shared parameters, while preserving one-step ensemble generation.

How can auxiliary tasks help one-step forecasters learn the predictive distribution under high uncertainty?

We address this question with auxiliary tasks that predict the same future from the context and an auxiliary variable carrying partial information about the future. We hypothesize that learning these tasks can help a single forecaster better capture the context-future relationship, as this information reduces prediction ambiguity. To include the original forecasting problem, the auxiliary variable should admit an uninformative case and, ideally, allow control over how much future information it provides. Diffusion forward process (Sohl-Dickstein et al., 2015) naturally satisfies these requirements: the corrupted future becomes uninformative at the fully corrupted endpoint, and the noise level controls the amount of future information (Kingma et al., 2021). We therefore construct auxiliary conditional denoising tasks that predict the same clean future from its corrupted version and the forecast context.

To learn such conditional denoising tasks while retaining one-step forecasting, we introduce a proper scoring rule-based diffusion framework built on distributional diffusion models (DDMs; De Bortoli et al., 2025). In our framework, a stochastic predictor learns the conditional distributions of these tasks and generates ensemble members directly at the fully corrupted endpoint. With component-wise CRPS as the scoring rule, we show that CRPS training is an endpoint-only special case of our formulation. As illustrated in Figure 1, our framework can therefore extend existing CRPS-based weather forecasters by adding only the corrupted future and its noise level as conditioning inputs.

Through extensive experiments across diverse architectures and forecast horizons, we show that our formulation improves one-step forecasting over standard CRPS training, with larger gains under high uncertainty. These gains extend to high-dimensional global weather forecasting under both training from scratch and fine-tuning. We also observe improved generalization under distribution shift and in tropical cyclone cases, consistent with a possible regularizing effect of auxiliary tasks.

Our contributions are summarized below.

- We introduce a proper scoring rule-based diffusion framework that uses auxiliary conditional denoising tasks to learn predictive distributions more effectively under high uncertainty.
- We revisit standard CRPS training as the endpoint-only special case of our formulation and generalize it with the auxiliary conditional denoising tasks.
- Building on this connection, we analyze how auxiliary conditional denoising tasks can improve one-step forecasting and support our interpretation through extensive controlled experiments.
- We find that these benefits extend to high-dimensional weather forecasting across architectures and training regimes, with only minor conditioning changes to existing CRPS-based forecasters.

2 PRELIMINARIES

In this section, we briefly review probabilistic weather forecasting, focusing on scoring rule-based and diffusion-based approaches. Further background including related work is provided in Section A.

Probabilistic Weather Forecasting Let X^τ denote the atmospheric state at time step τ, with consecutive steps separated by a fixed forecast interval. Probabilistic weather forecasting commonly adopts an autoregressive formulation, factorizing the distribution over forecast trajectories into one-step conditional distributions (Price et al., 2025; Alet et al., 2025),

$$p\left(X^{1:T} \mid c^0\right) = \prod_{\tau=0}^{T-1} p\left(X^{\tau+1} \mid c^\tau\right), \tag{1}$$

where the forecast context c^τ consists of the current state X^τ and may additionally include previous states and static and forcing variables. For notational simplicity, we write $X = X^{\tau+1}$ and $c = c^\tau$, so the goal is to model the predictive distribution $p(X \mid c)$ (Gneiting & Katzfuss, 2014). In practice, a learned model $p_\theta(X \mid c)$ generates an ensemble of M future states $\{\hat{X}^{(m)}\}_{m=1}^M$ approximating this distribution, and each member is rolled out by updating the context with its predicted state.

Scoring Rules for Weather Forecasting Scoring rules assign a scalar loss to a predictive distribution and an observed outcome. Throughout this work, we adopt the convention that lower scores are better. They are *proper* if expected loss is minimized when the predictive distribution matches the true distribution, and *strictly proper* if the minimizer is unique (Gneiting & Raftery, 2007).

In weather forecasting, a common choice is CRPS (Matheson & Winkler, 1976), which is given by

$$S_{\text{CRPS}}\left(p_\theta(\cdot \mid c), X\right) = \mathbb{E}\left[\|\hat{X} - X\|_1\right] - \frac{1}{2}\mathbb{E}\left[\|\hat{X} - \hat{X}'\|_1\right], \tag{2}$$

where $\hat{X}, \hat{X}' \overset{\text{i.i.d.}}{\sim} p_\theta(\cdot \mid c)$ and $\|\cdot\|_1$ sums absolute differences over forecast channels and grid points. With M ensemble members, we estimate S_{CRPS} with the unbiased *fair CRPS* (Ferro, 2014),

$$\hat{S}_{\text{CRPS}}\left(\{\hat{X}^{(m)}\}_{m=1}^M, X\right) = \frac{1}{M}\sum_{m=1}^{M}\|\hat{X}^{(m)} - X\|_1 - \frac{1}{2M(M-1)}\sum_{m\neq n}\|\hat{X}^{(m)} - \hat{X}^{(n)}\|_1. \tag{3}$$

Beyond evaluation, CRPS can also serve as a training objective for learning predictive distributions. Weather forecasting models can directly learn $p_\theta(\cdot \mid c)$ by minimizing a CRPS-based ensemble loss and generate each ensemble member in a single forward pass (Alet et al., 2025; Lang et al., 2026).

Diffusion-based Weather Forecasting Diffusion-based forecasters model the predictive distribution $p(X \mid c)$ by iteratively denoising random noise ε into future states conditioned on the context.

Following Lipman et al. (2023), we consider the forward corruption using a linear noise schedule,

$$q_t(X_t \mid X) = \mathcal{N}\big(X_t \mid (1-t)X, t^2 I\big), \qquad X_t = (1-t)X + t\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \tag{4}$$

where $t \in [0, 1]$ is the noise level. Reverse sampling uses a denoiser $D_\theta(c, X_t, t)$ trained to minimize

$$\mathcal{L}_{\text{Diffusion}}(\theta) = \int_0^1 w_t\, \mathbb{E}_{c, X, X_t \sim q_t(\cdot|X)}\left[\|X - D_\theta(c, X_t, t)\|^2\right] \mathrm{d}t, \tag{5}$$

where $w_t \geq 0$ weights noise levels. The optimal denoiser outputs the posterior mean, $D^*(c, X_t, t) = \mathbb{E}[X \mid c, X_t, t]$. Within DDIM sampling (Song et al., 2021), the denoiser output serves as a point-estimate approximation of the clean-state posterior, $p(\cdot \mid c, X_t, t) \approx \delta_{D_\theta(c, X_t, t)}$, and this approximation can degrade sample quality under coarse time discretization (De Bortoli et al., 2025). Generating high-quality ensemble members therefore typically requires fine time discretization, with repeated denoiser evaluations for each member incurring substantial inference cost (Price et al., 2025).

3 PROPER SCORING RULE-BASED DIFFUSION FORECASTER

We first examine challenges in context-only forecasting, which learns the predictive distribution $p(X \mid c)$ of the future state X from the context c alone. Motivated by these challenges, we introduce auxiliary conditional denoising tasks and evaluate their benefits. We then build on distributional diffusion models (DDMs; De Bortoli et al., 2025) to formalize a proper scoring rule-based diffusion forecaster that retains one-step ensemble generation. Finally, we show that standard CRPS training can be viewed as an endpoint-only special case of our formulation and present the resulting algorithm.

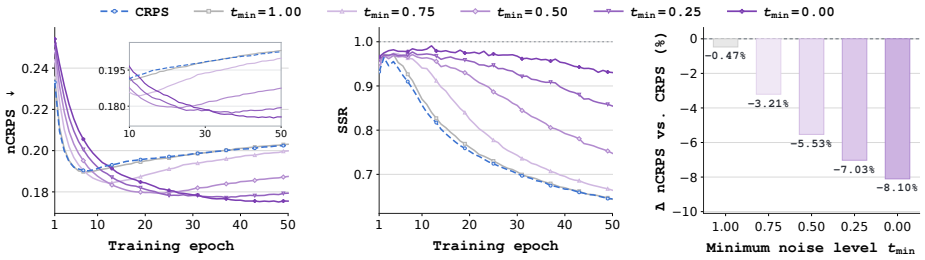


Figure 4: **Effect of conditional denoising tasks on one-step forecasting**. For 3-day-ahead weather forecasting, corrupted futures with noise levels $t \in [t_{\min}, 1]$ serve as additional training inputs. Panels show validation nCRPS (left), spread–skill ratio (SSR, middle), and test nCRPS change (%) relative to the CRPS-trained baseline (right). Endpoint-only training ($t_{\min} = 1$) closely matches the CRPS baseline. Lowering $t_{\min}$ broadens the range of auxiliary tasks, mitigating performance degradation.

3.1 Motivating Observations: Challenges of Context-Only Forecasting

To examine how context-only forecasting behaves as uncertainty increases, we directly forecast the weather 12 hours and 1, 3, and 5 days ahead without autoregressive rollout. As shown in Figure 2, the validation loss of the CRPS-trained models generally improves at 12 hours and 1 day. In contrast, at 3 and 5 days, it initially improves and then deteriorates while the training loss keeps decreasing, indicating overfitting. Similar patterns appear across diverse architectures, as shown in Figure 10. These observations suggest that context-only CRPS training can struggle at longer forecast horizons, prompting us to seek more effective ways to learn the same predictive distribution $p(X \mid c)$.

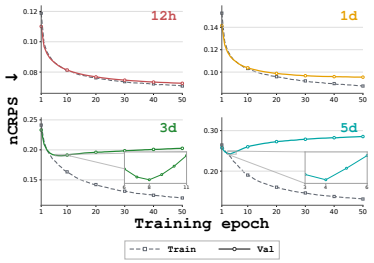


Figure 2: FCN3 (Bonev et al., 2025) train and validation normalized CRPS (nCRPS) across forecasting horizons.

Intuition of Auxiliary Denoising Tasks Motivated by evidence that joint training on related tasks can improve predictive performance on a target task (Caruana, 1997), we hypothesize that jointly learning to predict the same future X with access to partial information can improve context-only forecasting through shared parameters. As illustrated in Figure 3, partial future information provides clues about the future to be predicted and can reduce the ambiguity of the auxiliary prediction. Learning the same context-future relationship under these conditions may provide useful supervision for the shared model, helping address the difficulties observed in context-only forecasting. Shared training may also act as regularization and improve generalization in the original context-only forecasting task.

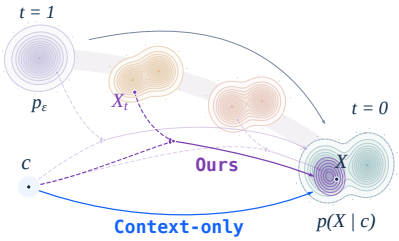


Figure 3: Schematic comparison of the context-only forecasting task and auxiliary conditional denoising tasks.

Under this intuition, we seek a family of auxiliary prediction tasks that includes the original forecasting problem $p(X \mid c)$ while allowing control over the amount of additional future information supplied to each task. The diffusion process in (4) naturally satisfies these requirements. Varying the noise level controls the information of X_t, and the independent-noise endpoint recovers context-only forecasting. This yields a family of conditional denoising tasks with target distributions $\{p(X \mid c, X_t, t)\}_{t\in[0,1]}$.

Effect of Auxiliary Denoising Tasks To assess the hypothesis, we jointly train a shared stochastic predictor on such conditional denoising tasks with the objective formalized in Section 3.2, building on DDMs (De Bortoli et al., 2025). We revisit the 3-day-ahead weather forecasting setting and vary the training noise range $t \in [t_{\min}, 1]$, while inference always uses a single forward pass at $t = 1$ (NFE $= 1$). As shown in Figure 4, endpoint-only training ($t_{\min} = 1$) closely mirrors the validation trajectories of the CRPS-trained, context-only baseline. Lowering $t_{\min}$ provides the model with richer auxiliary supervision, improving one-step forecasting and mitigating overfitting in line with our hypothesis. Additional experimental details and results are provided in Section D.2.

3.2 Proper Scoring Rule-based Diffusion for Auxiliary Denoising Tasks

We now formalize the joint learning of the conditional predictive distributions introduced in Section 3.1. Building on DDMs (De Bortoli et al., 2025), we adopt a proper scoring rule-based objective for a shared stochastic predictor and present its empirical estimator based on finite ensembles.

Proper Scoring Rule-based Diffusion To learn the family of conditional predictive distributions $\{p(X \mid c, X_t, t)\}_{t\in[0,1]}$ indexed by the noise level t, we minimize the expected scoring loss across noise levels using a shared stochastic predictor. Specifically, with the auxiliary variable $X_t = (1-t)X + t\varepsilon$ and $\varepsilon \sim \mathcal{N}(0, I)$ independent of (c, X) in (4), the training objective is

$$\mathcal{L}^{\lambda,\beta}_{\mathrm{DDM}}(\theta) = \int_0^1 w_t \, \mathbb{E}_{c,X,X_t\sim q_t(\cdot|X)} \left[S_{\lambda,\beta}\left(p_\theta(\cdot \mid c, X_t, t), X\right)\right] \mathrm{d}t, \tag{6}$$

where $p_\theta(\cdot \mid c, X_t, t)$ is the trainable predictive distribution. The nonnegative weight w_t controls the contribution of each noise level, and we use uniform weighting ($w_t \equiv 1$) throughout this work.

Here, the scoring rule $S_{\lambda,\beta}$ is a *generalized energy score* (De Bortoli et al., 2025) that provides a sample-based criterion for learning the conditional predictive distribution from observed targets,

$$S_{\lambda,\beta}(P, X) = \mathbb{E}_{\hat{X}\sim P}\left[\|\hat{X} - X\|^\beta\right] - \frac{\lambda}{2}\mathbb{E}_{(\hat{X},\hat{X}')\sim P\otimes P}\left[\|\hat{X} - \hat{X}'\|^\beta\right], \tag{7}$$

where $\lambda \in [0, 1]$ and $\beta \in (0, 2]$ control dispersion and distance sensitivity, respectively. With a proper scoring rule ($\lambda = 1$), a predictor satisfying $p_\theta(\cdot \mid c, X_t, t) = p(\cdot \mid c, X_t, t)$ for all t attains a population minimum of (6). At this solution, $p_\theta(\cdot \mid c, X_1, 1) = p(\cdot \mid c)$, so each ensemble member can be sampled from the target predictive distribution in a single forward pass, and multi-step sampling also recovers the same target distribution. In practice, learning an accurate one-step predictor can be challenging under high uncertainty, and multi-step sampling may improve forecasting in such cases.

Empirical Computation We estimate this objective using N training pairs (c_n, X_n) and $M \geq 2$ predicted samples $\{\hat{X}_n^{(m)}\}_{m=1}^M$ for each corrupted future state X_{t_n} with $t_n \sim \mathcal{U}(0, 1)$ as

$$\widehat{\mathcal{L}}^{\lambda,\beta}_{\mathrm{DDM}}(\theta) = \frac{1}{N}\sum_{n=1}^N \left[\frac{1}{M}\sum_{m=1}^M \left\|\hat{X}_n^{(m)} - X_n\right\|^\beta - \frac{\lambda}{2M(M-1)}\sum_{m\neq m'}\left\|\hat{X}_n^{(m)} - \hat{X}_n^{(m')}\right\|^\beta\right], \tag{8}$$

$$\zeta_n^{(m)} \overset{\text{i.i.d.}}{\sim} p_\zeta \xrightarrow{\text{introducing stochasticity}} \hat{X}_\theta(c_n, X_{t_n}, t_n, \zeta_n^{(m)}) = \hat{X}_n^{(m)} \sim p_\theta(\cdot \mid c_n, X_{t_n}, t_n). \tag{9}$$

Here, an additional stochastic source ζ, distinct from the corruption noise ε, allows the predictor to represent the predictive distribution by generating multiple clean futures under the same conditioning.

3.3 Revisiting CRPS Training from a Diffusion Perspective

We first establish standard context-only CRPS training as an endpoint-only special case of our formulation. We then present the resulting training algorithm, which extends standard CRPS training by incorporating auxiliary conditional denoising tasks while retaining one-step ensemble generation.

Generalization of CRPS Training From the distributional diffusion view in Section 3.2, standard CRPS training can be understood as an endpoint-only special case. If we set $\lambda = \beta = 1$ and apply the score component-wise, we obtain $S_{1,1} = S_{\mathrm{CRPS}}$. With training restricted to $t = 1$ and the endpoint predictive distribution independent of corruption noise, Equation 6 reduces to the CRPS objective[1]:

$$\begin{aligned}\mathcal{L}^{1,1}_{\mathrm{DDM}}(\theta) &= \int_0^1 \mathbb{E}_{c,X,\,X_t\sim q_t(\cdot|X)}\left[S_{\mathrm{CRPS}}\left(p_\theta(\cdot \mid c, X_t, t), X\right)\right] \mathrm{d}t \\ &\xrightarrow{\text{restrict training to } t=1} \mathbb{E}_{c,X}\left[S_{\mathrm{CRPS}}\left(p_\theta(\cdot \mid c), X\right)\right].\end{aligned} \tag{10}$$

From this view, standard CRPS training can be interpreted as learning only the endpoint of the task family in Section 3.1, and our formulation as generalizing it with auxiliary denoising tasks at $t < 1$.

[1] See Section B.2 for the mathematical details.

Training Algorithm In this work, we set $\lambda = \beta = 1$ and apply the score component-wise, yielding a CRPS-based DDM loss that naturally extends standard CRPS training with auxiliary denoising tasks across noise levels.

The training procedure is summarized in Algorithm 1, with *the only additional operations* relative to standard CRPS training highlighted in ***purple***. These operations sample a noise level t, construct a corrupted future X_t, and provide both as additional conditioning inputs. For each context–future pair (c, X), we generate M ensemble members $\{\hat{X}^{(m)}\}_{m=1}^{M}$ conditioned on the same corrupted future X_t, using independent draws of $\zeta^{(m)}$ to introduce stochasticity. At each sampled noise level, the predictor is then updated using a loss with the fair CRPS form.

Algorithm 1 Training

```
# model - stochastic predictor
# M - ensemble size (M >= 2)
for c, x in dataloader:
  # Corrupt the future
  t = uniform(0, 1)
  eps = randn_like(x)
  x_t = (1-t)*x + t*eps

  # Stochastic Ensemble Generation
  x_hat = []
  for m in range(M):
    zeta = sample(p_zeta)
    pred = model(c,zeta,x_t=x_t,t=t)
    x_hat.append(pred)

  # CRPS update
  loss = fair_crps(x_hat, x)
  update(model, loss)
```

The resulting training objective averages this fair CRPS over N context–future pairs and noise levels:

$$\hat{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{M}\sum_{m=1}^{M}\left\|\hat{X}_n^{(m)} - X_n\right\|_1 - \frac{1}{2M(M-1)}\sum_{m\neq m'}\left\|\hat{X}_n^{(m)} - \hat{X}_n^{(m')}\right\|_1\right], \tag{11}$$

where $\hat{X}_n^{(m)} = \hat{X}_\theta(c_n, X_{t_n}, t_n, \zeta_n^{(m)})$ and $\|\cdot\|_1$ sums absolute differences across forecast channels and spatial locations. We apply area and channel weighting as detailed in Appendix C, but omit them here for simplicity. One-step and multi-step sampling procedures are provided in Appendix B.1.

4 Experiments

We evaluate our formulation against standard CRPS training in two settings. In the controlled pilot setting, we vary the forecast horizon to assess the contribution of auxiliary conditional denoising tasks under different levels of uncertainty. In the high-dimensional global setting, we test whether the gains transfer across training regimes, including training from scratch and fine-tuning, and examine how the auxiliary tasks affect generalization under distribution shift and in tropical cyclone cases. Both settings include multiple architectures, allowing us to assess the consistency of the benefits.

Experimental Setup We use ERA5 reanalysis with `1979-2017`, `2018-2019`, and `2020` for training, validation, and testing, respectively (Hersbach et al., 2020). The controlled experiments use a $2.8125°$ (64×128) global latitude–longitude grid with 9 representative channels, while the large-scale experiments use a $1.5°$ (121×240) grid with 84 channels. We refer to models trained with the DDM objective as DDM and compare them against standard CRPS training under matched architectures and training configurations. All results in the main text use NFE $= 1$ unless stated otherwise. Further details including data preprocessing and additional results are described in Appendices C and D.

4.1 Pilot Experiments: Analysis of Auxiliary Conditional Denoising Tasks

In this $2.8125°$ setting, we consider two CRPS-based model architectures, FGN (Alet et al., 2025) and FCN3 (Bonev et al., 2025), with direct forecast horizons of **`12h`**, **`1d`**, **`3d`**, and **`5d`**. Each direct forecast horizon serves as the *rollout interval* (RI) during autoregressive rollouts. Detailed experimental configurations and additional results including other models are provided in Appendices C.4 and D.3.

Direct Forecasting Performance We first assess the contribution of auxiliary conditional denoising tasks to direct forecasting at NFE $= 1$ under different levels of uncertainty. As shown in Table 1, training with the DDM objective consistently yields better performance than standard CRPS training.

Specifically, the gains are larger in the **`3d`** and **`5d`** tasks, where predicting future states directly from context is more challenging, than for shorter forecast horizons. Under high uncertainty, richer supervision from auxiliary conditional denoising tasks may help the shared model better approximate the target predictive distribution. The noise-range ablation in Figure 4 also supports this interpretation, showing that broadening the range of auxiliary tasks improves NFE $= 1$ forecasting in the **`3d`** task.

Table 1: Direct Forecasting Performance

Model	Objective	nCRPS ↓ 12h	1d	3d	5d	nRMSE ↓ 12h	1d	3d	5d
FGN	CRPS	0.0581	0.0792	0.1868	0.2463	0.1138	0.1586	0.3847	0.5034
	DDM	0.0577	0.0778	0.1692	0.2290	0.1130	0.1559	0.3542	0.4760
	Δ (%)	-0.8	-1.8	-9.4	-7.0	-0.7	-1.7	-7.9	-5.4
FCN3	CRPS	0.0736	0.0967	0.1915	0.2452	0.1461	0.1937	0.3920	0.5029
	DDM	0.0735	0.0942	0.1759	0.2283	0.1461	0.1894	0.3664	0.4755
	Δ (%)	-0.1	-2.6	-8.1	-6.9	-0.0	-2.2	-6.6	-5.5

Table 2: Autoregressive Rollout Performance

Model	RI	Obj.	nCRPS ↓ 12h	72h	216h	nRMSE ↓ 12h	72h	216h
FGN	12h	CRPS	0.0581	0.1420	0.2391	0.1138	0.2986	0.4981
		DDM	0.0577	0.1419	0.2391	0.1130	0.2985	0.4985
		Δ (%)	-0.8	-0.1	-0.0	-0.7	-0.0	+0.1
	3d	CRPS	–	0.1868	0.2557	–	0.3847	0.5239
		DDM	–	0.1692	0.2451	–	0.3542	0.5088
		Δ (%)	–	-9.4	-4.1	–	-7.9	-2.9
FCN3	12h	CRPS	0.0736	0.1634	0.2474	0.1461	0.3399	0.5133
		DDM	0.0735	0.1632	0.2468	0.1461	0.3394	0.5119
		Δ (%)	-0.1	-0.1	-0.2	-0.0	-0.1	-0.3
	3d	CRPS	–	0.1915	0.2558	–	0.3920	0.5246
		DDM	–	0.1759	0.2461	–	0.3664	0.5118
		Δ (%)	–	-8.1	-3.8	–	-6.6	-2.4

Figure 5: Effect of NFE

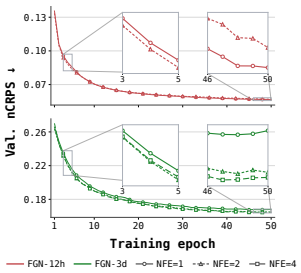
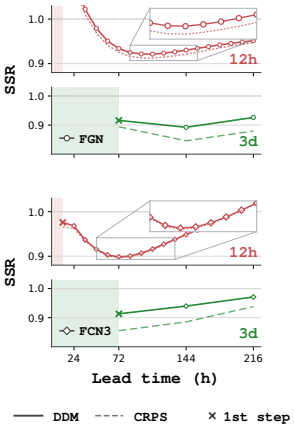


Figure 6: SSR Analysis



In contrast, gains are smaller in 12h and 1d. As noted earlier, both formulations admit population-optimal solutions that recover the true predictive distribution $p(X \mid c)$. In these easier tasks, CRPS training may already approximate the forecast marginals well, limiting potential gains from auxiliary conditional denoising. This provides one possible explanation for their similar performance.

Effect of Additional NFEs The benefits of additional NFEs can also be interpreted from the optimality perspective. At an ideal population optimum where the joint conditional denoising distributions are recovered exactly at all noise levels, DDM sampling recovers the true predictive distribution $p(X \mid c)$ regardless of NFE (see Section B.1). When one-step predictions already approximate this distribution well, additional NFEs may offer limited gains. Consistent with this expectation, Figure 5 shows that NFE gains in the 12h task are present only in early training. In contrast, gains persist in the 3d task, where learning an accurate one-step predictor may be more difficult and iterative refinement may improve the approximation of the target predictive distribution.

Autoregressive Rollout Performance We next evaluate autoregressive rollouts at NFE $= 1$ per rollout step. Across both architectures, DDM generally yields modest gains with a 12h rollout interval (RI) and maintains a clear performance advantage with a 3d interval (Table 2). At longer RIs, DDM also alleviates the underdispersion observed with CRPS training, maintaining SSRs closer to 1 and suggesting better ensemble calibration (Figure 6). Together, the pilot results support our expectation that auxiliary conditional denoising tasks improve direct forecasting at NFE $= 1$, with gains that persist through autoregressive rollouts and are more pronounced at longer RIs.

4.2 Scaling to High-Dimensional Global Weather Forecasting

Towards realistic global weather forecasting, we scale our evaluation to 1.5° ERA5 reanalysis with 84 forecast channels. We consider two recently proposed forecasting models, MOSAIC (Zhdanov et al., 2026) and U-Cast (Cachay et al., 2026), adapting their original training recipes. The two models provide complementary settings. MOSAIC is trained probabilistically from scratch and predicts the atmospheric state directly, whereas U-Cast is probabilistically fine-tuned from a model pretrained with mean absolute error (MAE) and predicts residuals. They further differ in how stochasticity

Table 3: CRPS and ensemble-mean RMSE across representative variables and forecast lead times[2].

	CRPS ↓									RMSE ↓								
	1d			3d			10d			1d			3d			10d		
Method	Z500	T850	Q700	Z500	T850	Q700	Z500	T850	Q700	Z500	T850	Q700	Z500	T850	Q700	Z500	T850	Q700
IFS-ENS†	24.1	0.370	0.283	59.3	0.550	0.427	263.8	1.343	0.725	45.8	0.718	0.611	133.4	1.105	0.902	623.1	2.819	1.440
GenCast†	20.2	0.269	0.222	54.3	0.459	0.367	253.6	1.281	0.676	39.3	0.542	0.498	123.5	0.955	0.807	606.5	2.737	1.375
NeuralGCM†	22.9	0.333	0.263	54.9	0.497	0.380	253.4	1.293	0.679	44.0	0.658	0.543	126.2	1.027	0.812	606.8	2.756	1.374
MOSAIC	22.7	0.300	0.239	58.3	0.480	0.372	261.0	1.303	0.680	44.2	0.601	0.525	133.2	1.001	0.814	619.9	2.781	1.383
$\text{MOSAIC}_{\text{DDM}}$	21.9	0.290	0.232	56.7	0.467	0.365	257.6	1.290	0.675	42.4	0.581	0.513	128.6	0.973	0.800	610.1	2.755	1.374
U-Cast	20.4	0.281	0.229	56.1	0.480	0.373	260.1	1.326	0.704	39.8	0.563	0.507	126.9	0.985	0.817	615.7	2.792	1.427
$\text{U-Cast}_{\text{DDM}}$	20.4	0.280	0.229	56.1	0.475	0.372	259.5	1.314	0.694	39.7	0.562	0.507	126.6	0.979	0.814	613.2	2.779	1.409

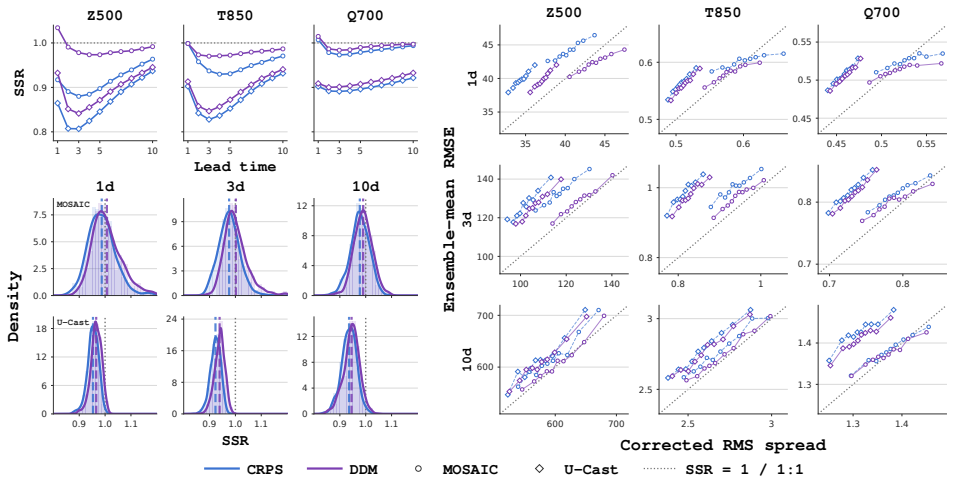


Figure 8: Ensemble calibration analysis. (Top left) SSR versus forecast lead time for Z500, T850, and Q700. (Bottom left) SSR distributions across all 84 forecast channels at 1-, 3-, and 10-day lead times. (Right) Binned spread–skill plots for the same variables at 1-, 3-, and 10-day lead times.

is introduced, allowing us to evaluate DDM across distinct training regimes and parameterizations. Detailed descriptions of the model designs and training configurations are provided in Appendix C.5.

Forecast Performance Table 3 compares CRPS and ensemble-mean RMSE on three variables at 1-, 3-, and 10-day lead times. At NFE $= 1$ per rollout step, DDM improves both metrics in all cases for MOSAIC and improves or matches them for U-Cast.

These improvements extend beyond the representative variables. Across all 84 forecast channels, DDM achieves lower CRPS in 787 of 840 channel–lead pairs (93.7%) for MOSAIC, as shown in Figure 7. For U-Cast, where the objectives differ only during brief fine-tuning, the gains are modest but extend across most channel–lead pairs (82.0%, see Figure 13). Together, these results indicate that auxiliary conditional denoising tasks benefit high-dimensional forecasting in both training from scratch and fine-tuning.

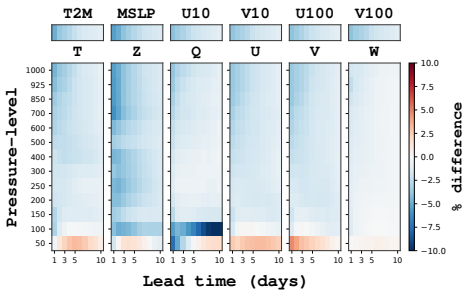


Figure 7: Relative CRPS difference between DDM- and CRPS-trained MOSAIC across all 84 forecast channels over 10-day rollouts. Blue indicates better performance of DDM.

Calibration Analysis We next examine whether the forecast-skill gains from DDM are accompanied by improved ensemble calibration. The bottom-left panels of Figure 8 show SSR distributions across all 84 forecast channels at 1-, 3-, and 10-day lead times. DDM shifts these distributions toward 1 across both architectures and the evaluated lead times, suggesting improved ensemble calibration.

[2]Results marked with † are provided for reference, as their configurations differ from our controlled experimental setup. Our primary comparisons evaluate standard CRPS and DDM training under matched settings.

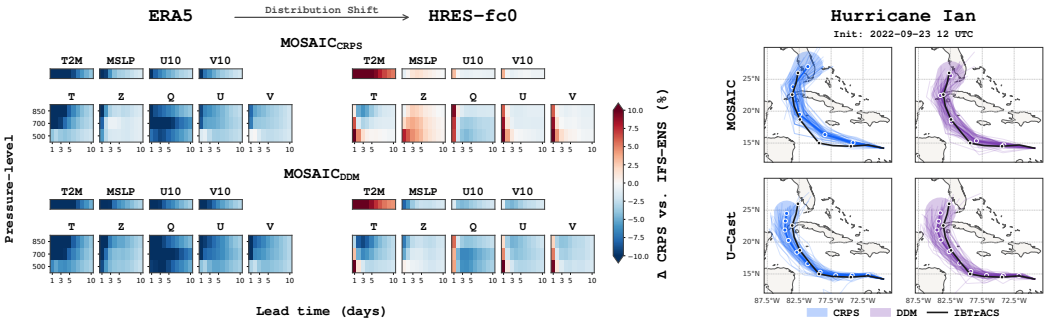


Figure 9: (Left) Relative CRPS differences from IFS-ENS for ERA5-trained MOSAIC evaluated on ERA5 and HRES-fc0. (Right) Ensemble track forecasts and IBTrACS best track for Hurricane Ian.

Binned spread–skill plots assess whether forecasts with larger ensemble spread have larger errors (Wang & Bishop, 2003; Delle Monache et al., 2013). Forecasts are binned by ensemble spread. Within each bin, we compare finite-ensemble-corrected RMS spread with the ensemble-mean RMSE. Proximity to the 1:1 line indicates agreement between ensemble spread and forecast error across levels of predicted uncertainty, and DDM generally brings the curves closer to this line (Figure 8).

4.3 Generalization under Distribution Shift and Extreme Weather

In Section 3.1, we show that introducing auxiliary conditional denoising mitigates the deterioration in validation performance late in training (Figure 4). Motivated by this observation and the potential regularizing effect of shared learning briefly discussed in Section 3.1, we assess the generalization ability of DDM-trained models under distribution shift and in tropical cyclone case studies.

Robustness to Distribution Shift We first evaluate ERA5-trained MOSAIC models on HRES-fc0 without fine-tuning, using the same 732 initialization times in 2020. Across all the channels and 1–10-day lead times, both models deteriorate relative to ERA5 evaluation, but DDM shows smaller average increases in nCRPS, nRMSE, and mean SSR deviation, as shown in Table 4. To assess the impact of this degradation, we further compare both models with IFS-ENS across 19 shared channels at the same lead times. On ERA5, CRPS- and DDM-trained MOSAIC achieve lower CRPS than IFS-ENS in 99.5% and 100% of these 190 channel–lead pairs, respectively. On HRES-fc0, this fraction drops to 58.4% for CRPS-trained MOSAIC, while DDM-trained MOSAIC still achieves lower CRPS than IFS-ENS in 86.8% of pairs. Full details are provided in Section D.4.2.

Table 4: Changes in performance under the ERA5-to-HRES distribution shift.

	nCRPS ↓	nRMSE ↓	SSR dev. ↓
Objective	Relative (%)	Relative (%)	Absolute
CRPS	+3.10	+2.68	+0.017
DDM	+2.60	+2.24	+0.007

Tropical Cyclone Tracking We also evaluate 50-member ensemble forecasts for Hurricane Ian (2022) as an extreme-weather case study. We track cyclone centers following Zhdanov et al. (2026) and compare the predicted tracks with the IBTrACS best track (Knapp et al., 2010). Figure 9 shows predicted tracks and uncertainty cones enclosing 66% of the ensemble track distribution. In this case, DDM produces track distributions closer to the IBTrACS best track for both architectures. Additional results including Hurricane Michael and quantitative tracking metrics are provided in Section D.4.3.

5 Conclusion

We introduced a proper scoring rule-based diffusion framework that incorporates auxiliary conditional denoising tasks, with standard CRPS training recovered as an endpoint-only special case. Through controlled experiments, we found that jointly learning the auxiliary tasks improves one-step forecasting performance and mitigates overfitting, particularly under high uncertainty. These benefits also extend to high-dimensional global weather forecasting across different architectures and training regimes, while evaluations under distribution shift and in tropical cyclone cases suggest potential generalization benefits. Our framework achieves this with only minor conditioning changes to existing CRPS-based forecasters while retaining one-step ensemble generation. However, our work does not establish theoretical guarantees for these benefits, leaving an interesting direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to support qualitative and thematic data analysis, interpret results, implement methods, provide feedback on research methodology and experiments, refine hypotheses, formulate mathematical claims, provide critical ingredients for proving mathematical claims, and assist in writing proofs. We have not used generative AI tools to originate the core ideas or formulation, help develop theoretical models or conceptual frameworks, assist with translation, or clean and reformat datasets, and synthetic dataset generation is not applicable to this work. Additionally, we used generative AI tools to create artifacts, create or modify scientific figures, draft parts of the manuscript, search for information, edit the manuscript to improve readability, and suggest a structure for the manuscript. We have reviewed all AI-assisted work. The authors independently developed the core ideas and formulation, established the overall proof strategies and experimental design, and prepared the initial figure designs before seeking AI assistance. All LLM-generated code was reviewed by the authors for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies probabilistic weather forecasting using publicly available meteorological data and does not involve human participants or personal data. Improved ensemble forecasting may support applications such as disaster preparedness and renewable energy planning. However, forecast errors and miscalibrated uncertainty estimates could adversely affect decisions based on these forecasts. Operational use therefore requires further validation.

REPRODUCIBILITY STATEMENT

The training objective and algorithm are described in Section 3.3 and Algorithm 1, and the sampling procedures in Appendix B.1. Appendix C specifies the data splits, preprocessing, normalization, loss weighting, evaluation metrics, architectural modifications, and training configurations for all experiments. The details of noise priors and schedules are provided in Appendix D.3.2.

REFERENCES

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. URL https://arxiv.org/abs/2506.10772v1.

Martin Andrae, Tomas Landelius, Joel Oskarsson, and Fredrik Lindsten. Continuous ensemble weather forecasting with diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

Kendall Atkinson and Weimin Han. *Spherical harmonics and approximations on the unit sphere: an introduction*, volume 2044 of *Lecture Notes in Mathematics*. Springer Heidelberg, 2012.

Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Boris Bonev, Thorsten Kurth, Ankur Mahesh, Mauro Bisson, Jean Kossaifi, Karthik Kashinath, Anima Anandkumar, William D Collins, Michael S Pritchard, and Alexander Keller. FourCastNet 3: A geometric approach to probabilistic machine-learning weather forecasting at scale, 2025. URL https://arxiv.org/abs/2507.12144v2.

Salva Rühling Cachay, Duncan Watson-Parris, and Rose Yu. U-Cast: A surprisingly simple and efficient frontier probabilistic AI weather forecaster. In *International Conference on Machine Learning (ICML)*, 2026.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. *npj climate and atmospheric science*, 6(1):190, 2023.

Guillaume Couairon, Renu Singh, Anastase Charantonis, Christian Lessig, and Claire Monteleoni. ArchesWeatherGen: Skillful and compute-efficient probabilistic weather forecasting with machine learning. *Science Advances*, 12(17):eadx2372, 2026.

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 *International Conference on Machine Learning (ICML)*, 2025.

Luca Delle Monache, F Anthony Eckel, Daran L Rife, Badrinath Nagarajan, and Keith Searight. Probabilistic weather prediction with an analog ensemble. *Monthly Weather Review*, 141(10): 3498–3516, 2013.

Christopher AT Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014.

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

Milton Gomez, Marie McGraw, Saranya Ganesh S., Frederick Iat-Hin Tam, Ilia Azizi, Samuel Darmon, Monika Feldmann, Stella Bourdin, Louis Poulain-Auzéau, Suzana J. Camargo, Jonathan Lin, Dan Chavas, Chia-Ying Lee, Ritwik Gupta, Andrea Jenney, and Tom Beucler. TCBench: A benchmark for tropical cyclone track and intensity forecasting at the global scale, 2026. URL https://arxiv.org/abs/2601.23268v2.

Ian Goodfellow, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Multi-prediction deep boltzmann machines. In *Conference on Neural Information Processing Systems (NIPS)*, 2013.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

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

Emiel Hoogeboom and Tim Salimans. Blurring diffusion models. In *International Conference on Learning Representations (ICLR)*, 2023.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Kenneth R Knapp, Michael C Kruk, David H Levinson, Howard J Diamond, and Charles J Neumann. The international best track archive for climate stewardship (IBTrACS): Unifying tropical cyclone data. *Bulletin of the American Meteorological Society*, 91(3):363–376, 2010.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, Sam Hatfield, Peter Battaglia, Alvaro Sanchez-Gonzalez, Matthew Willson, Michael P. Brenner, and Stephan Hoyer. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024.

Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. FourCastNet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Platform for Advanced Scientific Computing Conference (PASC)*, 2023.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677): 1416–1421, 2023.

Simon Lang, Mihai Alexe, Mariana C. A. Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D. Dueben, Sara Hahner, Pedro Maciel, Ana Prieto-Nemesio, Cathal O'Brien, Florian Pinault, Jan Polster, Baudouin Raoult, Steffen Tietsche, and Martin Leutbecher. AIFS-CRPS: ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *npj Artificial Intelligence*, 2(1):18, 2026.

Martin Leutbecher and Tim N Palmer. Ensemble forecasting. *Journal of computational physics*, 227 (7):3515–3539, 2008.

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

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision (ICCV)*, 2021.

James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research (JMLR)*, 17(81):1–32, 2016.

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. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems (JAMES)*, 16(6):e2023MS004019, 2024.

Stephan Rasp, Boris Babenko, Dominic Masters, Andrew El-Kadi, Samier Merchant, Guy Shalev, Ilan Price, Fred Zyda, Remi Lam, Sasha Shysheya, Matthew Willson, Stratis Markou, Shreya Agrawal, Suhani Vora, Mohammed Alewi Hassen, Sunny Mak, Tom R. Andersson, Megan Bela, Akib Uddin, Nofar Peled Levi, Ben Gaiarin, Ferran Alet, Aaron Bell, Peter Battaglia, and Alvaro Sanchez-Gonzalez. WeatherNext 3: Increasing resolution and performance of global weather models with raw observations, 2026. URL https://arxiv.org/abs/2609.03582v1.

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

Julia Slingo and Tim Palmer. Uncertainty in weather and climate prediction. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 369(1956):4751–4767, 2011.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Jason Stock, Troy Arcomano, and Rao Kotamarthi. Swift: An autoregressive consistency model for efficient weather forecasting. *Machine Learning: Earth*, 2(2):025004, 2026.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2008.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research (JMLR)*, 11(110):3371–3408, 2010.

Xuguang Wang and Craig H Bishop. A comparison of breeding and ensemble transform kalman filter ensemble forecast schemes. *Journal of the atmospheric sciences*, 60(9):1140–1158, 2003.

Fan Xu, Yuan Gao, Kun Wang, Rui Su, Fenghua Ling, Hao Wu, and Wanli Ouyang. Tyche: One step flow for efficient probabilistic weather forecasting, 2026. URL https://arxiv.org/abs/2605.06916v1.

Maksim Zhdanov, Ana Lucic, Max Welling, and Jan-Willem van de Meent. (sparse) attention to the details: Preserving spectral fidelity in ML-based weather forecasting models. In *International Conference on Machine Learning (ICML)*, 2026.

A Related Work

Scoring Rules for Probabilistic Weather Forecasting Proper scoring rules serve as both evaluation metrics and training objectives for probabilistic forecasting (Gneiting & Raftery, 2007). For global weather forecasting, AIFS-CRPS, FGN, and FourCastNet3 (FCN3) train stochastic predictors with CRPS-based objectives and generate samples without iterative denoising (Lang et al., 2026; Alet et al., 2025; Bonev et al., 2025). MOSAIC addresses spectral fidelity through its architecture and output parameterization (Zhdanov et al., 2026), while U-Cast combines deterministic pretraining with CRPS-based fine-tuning and Monte Carlo dropout (Cachay et al., 2026). WeatherNext 3 extends FGN to incorporate observations and predict multiple data modalities (Rasp et al., 2026).

Diffusion-based Weather Forecasting GenCast and ArchesWeatherGen generate ensemble members with sharp spatial features through iterative diffusion and flow-based sampling, respectively (Price et al., 2025; Couairon et al., 2026). Repeated model evaluations at each forecast step make generating large ensembles over long rollouts computationally expensive. Efficiency can therefore be improved by parallelizing forecast lead times or reducing the sampling cost of each forecast step. For the former, Andrae et al. (2025) combine lead-time conditioning with correlated noise to generate temporally coherent ensemble trajectories in parallel. For the latter, Swift (Stock et al., 2026) and Tyche (Xu et al., 2026) enable one-step generation through consistency and MeanFlow-based training, respectively, followed by autoregressive CRPS fine-tuning. These methods learn a deterministic one-step transport from noise and introduce CRPS in a separate fine-tuning stage. In contrast, our formulation applies the scoring rule at every noise level throughout training and requires only additional conditioning inputs to existing CRPS-based forecasters.

Auxiliary Denoising for One-step Forecasting Distributional diffusion models learn conditional distributions of clean data given corrupted observations across noise levels (De Bortoli et al., 2025). In our forecasting formulation, these prediction tasks retain the same future-state target while varying the information supplied by its corrupted version. Multi-step sampling uses intermediate denoising predictions explicitly, whereas our primary one-step setting evaluates the predictor only at the fully corrupted endpoint. In the latter setting, intermediate denoising tasks therefore provide auxiliary supervision through shared parameters rather than additional inference steps. This interpretation connects to denoising autoencoders, which demonstrate that reconstruction from corrupted inputs can support useful representation learning (Vincent et al., 2008; 2010), and to multitask learning, which exploits shared representations across related tasks (Caruana, 1997; Goodfellow et al., 2013; Maurer et al., 2016). Together, these perspectives motivate the hypothesis that learning intermediate denoising conditionals can benefit endpoint forecasting through shared learning, potentially acting as an implicit regularizer even when intermediate predictions are not used at inference.

B Mathematical Details

B.1 Sampling Procedures

In this work, we primarily use one-step sampling (NFE $= 1$). To explore the potential benefits of additional sampling steps, we follow the procedure described in De Bortoli et al. (2025).

One-step Sampling Given forecast context c, we generate M ensemble members using the stochastic predictor in (9). For each member, we independently draw corruption noise $\varepsilon^{(m)}$ from the endpoint distribution of the forward process and stochastic source $\zeta_0^{(m)} \sim p_\zeta$. The forecast is obtained through

$$\hat{X}^{(m)} = \hat{X}_\theta\left(c, \varepsilon^{(m)}, 1, \zeta_0^{(m)}\right), \qquad m = 1, \dots, M. \tag{12}$$

All members share the same context c but use independent draws of ε and ζ.

Multi-step Sampling For K-step sampling, we use the uniform grid $t_k = 1 - k/K$ and initialize $X_{t_0}^{(m)} = \varepsilon^{(m)}$. At each step, we independently draw $\zeta_k^{(m)} \sim p_\zeta$ and predict a clean future

$$\hat{X}_k^{(m)} = \hat{X}_\theta\left(c, X_{t_k}^{(m)}, t_k, \zeta_k^{(m)}\right). \tag{13}$$

For the linear noise schedule in Section 3, the coupled update is

$$X_{t_{k+1}}^{(m)} = \hat{X}_k^{(m)} + \frac{t_{k+1}}{t_k}\left(X_{t_k}^{(m)} - \hat{X}_k^{(m)}\right), \qquad k = 0, \dots, K-1. \tag{14}$$

The context c remains fixed throughout sampling and the stochastic source is resampled independently across members and steps. No additional corruption noise is injected between steps, but sampling remains stochastic through $\zeta_k^{(m)}$. The final forecast equals the last clean prediction, $X_{t_K}^{(m)} = \hat{X}_{K-1}^{(m)}$.

For the heat kernel schedule in Section D.3.2, the update in (14) is applied to each spherical-harmonic coefficient with its degree dependent noise level $t_\ell(\cdot)$ from (30). Omitting the member index,

$$(X_{t_{k+1}})_{\ell m} = (\hat{X}_k)_{\ell m} + \frac{t_\ell(t_{k+1})}{t_\ell(t_k)}\big((X_{t_k})_{\ell m} - (\hat{X}_k)_{\ell m}\big). \tag{15}$$

Exactness under Conditional Distribution Recovery Let $q_t(\cdot \mid c)$ denote the conditional law of $X_t = (1-t)X + t\varepsilon$, with ε independent of (c, X). For any finite grid $1 = t_0 > t_1 > \cdots > t_K = 0$, assume that the stochastic predictor recovers the true joint conditional distributions,

$$p_\theta(\cdot \mid c, x, t_k) = p(\cdot \mid c, x, t_k), \qquad k = 0, \dots, K-1, \tag{16}$$

where equality holds for almost every context c and $q_{t_k}(\cdot \mid c)$-almost every x.

For consecutive grid points $t = t_k > s = t_{k+1}$, let $Y_t \sim q_t(\cdot \mid c)$ denote the current sampler state and $\hat{X} \sim p_\theta(\cdot \mid c, Y_t, t)$ its clean prediction. The update is $Y_s = (1 - s/t)\hat{X} + (s/t)Y_t$ as written above. Exact conditional sampling restores the joint law of (X, X_t), and applying this update gives

$$\begin{aligned} \mathrm{Law}(\hat{X}, Y_t \mid c) &= \mathrm{Law}(X, X_t \mid c), \\ \mathrm{Law}(Y_s \mid c) &= \mathrm{Law}\left((1 - s/t)X + (s/t)X_t \mid c\right) \\ &= \mathrm{Law}\left((1-s)X + s\varepsilon \mid c\right) = q_s(\cdot \mid c). \end{aligned} \tag{17}$$

Starting from $q_1(\cdot \mid c)$, induction therefore yields the correct state distribution at every grid point and the final distribution $q_0(\cdot \mid c) = p(\cdot \mid c)$. This holds for any finite $K \geq 1$, including one-step sampling, and any corruption-noise prior independent of (c, X). The same argument applies coefficient-wise to the HK schedule, with s/t replaced by $t_\ell(s)/t_\ell(t)$. Therefore, under exact joint conditional recovery, DDM sampling recovers the target predictive distribution regardless of NFE.

B.2 CRPS at the Distributional Diffusion Endpoint

We consider component-wise CRPS with $\lambda = \beta = 1$ and assume finite first moments and finite expected scores. At the endpoint, $X_1 = \varepsilon$ is independent of (c, X), so $p(\cdot \mid c, X_1, 1) = p(\cdot \mid c)$.

Endpoint-only training places all probability mass at $t = 1$. If the predictive distribution satisfies $p_\theta(\cdot \mid c, \varepsilon, 1) = p_\theta(\cdot \mid c)$ for almost every (c, ε), its objective reduces to standard CRPS training,

$$\begin{aligned} \mathcal{L}_{\mathrm{end}}(\theta) &= \mathbb{E}_{c,X,\varepsilon}\left[S_{\mathrm{CRPS}}\left(p_\theta(\cdot \mid c, \varepsilon, 1), X\right)\right] \\ &= \mathbb{E}_{c,X}\left[S_{\mathrm{CRPS}}\left(p_\theta(\cdot \mid c), X\right)\right]. \end{aligned} \tag{18}$$

The first equality defines the endpoint-only objective, and the second uses independence from corruption noise. This restricts the predictive distribution and allows stochastic predictions through ζ.

If the predictive distribution depends on the corruption noise ε, the objectives need not coincide. To show that they nevertheless share the same population-optimal forecast marginals when optimized over all predictive distributions, let $F_{\theta,j}(u \mid c, \varepsilon)$ and $F_j(u \mid c)$ denote the endpoint predictive and target CDFs evaluated at $u \in \mathbb{R}$ for each component $j \in \{1, \dots, d\}$, respectively. The CDF representation of CRPS (Gneiting & Raftery, 2007) yields the population excess risk

$$\mathcal{L}_{\mathrm{end}}(\theta) - \mathcal{L}^* = \mathbb{E}_{c,\varepsilon}\left[\sum_{j=1}^{d}\int_{\mathbb{R}}\left(F_{\theta,j}(u \mid c, \varepsilon) - F_j(u \mid c)\right)^2 \mathrm{d}u\right], \tag{19}$$

where $\mathcal{L}^* = \mathbb{E}_{c,X}[S_{\mathrm{CRPS}}(p(\cdot \mid c), X)]$ is independent of θ. The nonnegative excess risk vanishes exactly when all predictive marginals match their targets for almost every (c, ε). Therefore, the true conditional forecast marginals minimize both the endpoint-only DDM and standard CRPS population objectives, even when fixed nonnegative channel and area weights are applied in Section C.1.

C Experimental Details

C.1 Data and Preprocessing

Data and Splits We conduct experiments at two spatial resolutions: 2.8125° on a 64×128 grid for the pilot experiments, and 1.5° on a 121×240 grid for the high-dimensional experiments. Both settings use 6-hourly ERA5 data, with `1979–2017` for training, `2018–2019` for validation, and `2020` for testing. For checkpoint and configuration selection, we compute validation nCRPS on a fixed set of 240 initializations evenly spaced over the validation period and shared across all runs.

Input and Target Normalization Let k index channels, X denote the future state, and X^{cur} denote the current state in the forecast context c. Dynamic input states are standardized channel-wise using the training-period mean μ_k and standard deviation σ_k. Target normalization depends on the prediction parameterization. FCN3 (Bonev et al., 2025) and MOSAIC (Zhdanov et al., 2026) predict the future state, while FGN (Alet et al., 2025) and U-Cast (Cachay et al., 2026) predict its residual relative to the current state. For each forecast lead, the normalized target is

$$T_k = \begin{cases} (X_k - \mu_k)/\sigma_k, & \text{state prediction,} \\ (X_k - X_k^{\mathrm{cur}})/s_k, & \text{residual prediction,} \end{cases} \tag{20}$$

where s_k is the training-period standard deviation of $X_k - X_k^{\mathrm{cur}}$ for the corresponding forecast lead.

Spatial and Channel Weighting All spatially aggregated metrics use cosine-latitude weights to account for unequal grid-cell areas. Let H and W denote the numbers of latitude and longitude grid points, respectively, and ϕ_h the latitude of row h. We normalize the area weights to have unit mean:

$$a_h = \frac{\cos(\phi_h)}{H^{-1}\sum_{h'=1}^{H}\cos(\phi_{h'})}. \tag{21}$$

Then, the spatially and channel-averaged training loss is

$$\mathcal{L} = \frac{1}{CHW}\sum_{k=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\gamma_k\, a_h\, \ell_{khw}, \tag{22}$$

where ℓ_{khw} is the grid-point loss and γ_k is the channel weight specified below.

- In the 2.8125° setting, we use nine representative channels: `Z500`, `T850`, `Q700`, `U850`, `V850`, `T2M`, `U10`, `V10`, and `MSLP`. All channels are weighted uniformly ($\gamma_k = 1$).
- In the 1.5° setting, we use the 84 dynamic state channels listed in Table 5, comprising 6 upper-air variables at 13 pressure levels and 6 surface or near-surface fields. Following the pressure-level weighting convention of Lam et al. (2023), each upper-air channel is assigned $\gamma_k = p/\bar{p}$, where $\bar{p} \approx 463.46$ hPa is the mean of the 13 pressure levels. For surface and near-surface fields, we use $\gamma_k = 1.0$ for $T2m$ and $\gamma_k = 0.1$ for all remaining channels, as summarized in Table 6.

C.2 Evaluation Metrics

We evaluate N initializations indexed by i. For initialization i, a forecast lead, and channel k, let $\{\hat{X}_i^{(m)}\}_{m=1}^{M}$ denote the ensemble forecasts with $M \geq 2$, $\bar{X}_i = M^{-1}\sum_{m=1}^{M}\hat{X}_i^{(m)}$ their mean, and X_i the verifying state. We omit the lead index throughout and the channel index k when unambiguous. Predictions are evaluated in state space after reconstructing states, and we use the normalized cosine-latitude weights a_h defined in Section C.1.

CRPS We average the fair CRPS estimates over initializations and grid points with area weights,

$$\mathrm{CRPS} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}a_h\left[\frac{1}{M}\sum_{m=1}^{M}\left|\hat{X}_{i,h,w}^{(m)} - X_{i,h,w}\right| - \frac{1}{2M(M-1)}\sum_{m\neq m'}\left|\hat{X}_{i,h,w}^{(m)} - \hat{X}_{i,h,w}^{(m')}\right|\right]. \tag{23}$$

Table 5: Channels used in the 1.5° experiments.

Variable	Unit	Channels
Surface and near-surface fields		
2-m temperature	K	1
Mean sea-level pressure	Pa	1
10-m zonal wind	$\mathrm{m\,s^{-1}}$	1
10-m meridional wind	$\mathrm{m\,s^{-1}}$	1
100-m zonal wind	$\mathrm{m\,s^{-1}}$	1
100-m meridional wind	$\mathrm{m\,s^{-1}}$	1
Upper-air fields		
Geopotential	$\mathrm{m^2\,s^{-2}}$	13
Temperature	K	13
Specific humidity	$\mathrm{kg\,kg^{-1}}$	13
Zonal wind	$\mathrm{m\,s^{-1}}$	13
Meridional wind	$\mathrm{m\,s^{-1}}$	13
Vertical velocity	$\mathrm{Pa\,s^{-1}}$	13
Total forecast channels		**84**
Static and forcing inputs (not predicted)		
Surface geopotential	$\mathrm{m^2\,s^{-2}}$	1
Land–sea mask	–	1
Latitude (sin)	–	1
Longitude (sin, cos)	–	2
Local time of day (sin, cos)	–	6
Time of year (sin, cos)	–	6
Total static and forcing channels		**17**

Table 6: Channel weights in 1.5° experiments.

Variable / Pressure level	Weight γ_k
Surface and near-surface variables	
2-meter temperature	1.0
10-meter zonal wind	0.1
10-meter meridional wind	0.1
100-meter zonal wind	0.1
100-meter meridional wind	0.1
Mean sea-level pressure	0.1
Pressure levels (all upper-air variables)	
1000 hPa	2.158
925 hPa	1.996
850 hPa	1.834
700 hPa	1.510
600 hPa	1.295
500 hPa	1.079
400 hPa	0.863
300 hPa	0.647
250 hPa	0.539
200 hPa	0.432
150 hPa	0.324
100 hPa	0.216
50 hPa	0.108

Ensemble-mean RMSE Following WeatherBench 2 (Rasp et al., 2024), we average the area-weighted squared error of the ensemble mean over initializations before taking the square root,

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} a_h\left(\bar{X}_{i,h,w} - X_{i,h,w}\right)^2}. \tag{24}$$

Spread–skill Ratio (SSR) For each initialization, we compute the unbiased ensemble variance

$$v_{i,h,w} = \frac{1}{M-1}\sum_{m=1}^{M}\left(\hat{X}^{(m)}_{i,h,w} - \bar{X}_{i,h,w}\right)^2 \tag{25}$$

and the corresponding spatial RMS spread

$$\mathrm{Spread}_i = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} a_h v_{i,h,w}}. \tag{26}$$

As for RMSE, we pool squared spread over initializations before taking the square root. Following Bonev et al. (2025), the finite-ensemble-corrected SSR for channel k is

$$\mathrm{SSR}_k = \sqrt{\frac{M+1}{M}}\,\frac{\sqrt{N^{-1}\sum_{i=1}^{N}\mathrm{Spread}_{i,k}^2}}{\mathrm{RMSE}_k}. \tag{27}$$

The correction accounts for the sampling variability of the finite ensemble mean. When reporting an aggregate across channels, we average the channel-wise SSRs.

Normalized CRPS and Ensemble-mean RMSE Let CRPS_k and RMSE_k denote the metrics above for channel k. Their normalized aggregates are

$$\mathrm{nCRPS} = \frac{1}{C}\sum_{k=1}^{C}\frac{\mathrm{CRPS}_k}{\sigma_k}, \qquad \mathrm{nRMSE} = \frac{1}{C}\sum_{k=1}^{C}\frac{\mathrm{RMSE}_k}{\sigma_k}. \tag{28}$$

Here σ_k is the training-period state normalization standard deviation. These summaries weight channels equally without the training loss weights γ_k. In the case of training curves in Figure 10, nCRPS uses the target normalization of each model, which may differ from the state normalization.

C.3 Architectural Modifications

Throughout this work, we adapt the forecasting backbones to DDM by adding conditioning on the corrupted target X_t and noise level t. These modifications retain the prediction parameterization and ensemble-generation mechanism of the corresponding CRPS-based weather forecasting models. Thus, the architectural changes are limited to expanding the input layers and adding time-conditioning modules, resulting in the modest parameter increases reported in Tables 7 and 9.

C.4 Experimental Details at 2.8125 degrees

Architectures The pilot study spans four backbones. Two are general-purpose architectures: Swin Transformer (**`Swin`**) (Liu et al., 2021), a hierarchical window-attention transformer, and **`U-Net`** (Ronneberger et al., 2015), a convolutional encoder–decoder with skip connections. The other two are CRPS-based probabilistic weather forecasting models: **`FGN`** (Alet et al., 2025), a graph-based model that injects noise through a low-dimensional latent vector, and **`FCN3`** (Bonev et al., 2025), a spherical neural operator that injects noise internally at multiple spatial scales. Each backbone is trained under both objectives with matched optimization settings.

Table 7: Backbones used in the $2.8125°$ pilot study.

			# Parameters		
Model	Target	Ensemble randomness	**`CRPS`**	**`DDM`**	Δ
`Swin`	residual	latent vector (d=32)	26.82M	26.84M	+0.09%
`U-Net`	residual	latent vector (d=32)	22.75M	22.76M	+0.05%
`FGN`	residual	latent vector (d=32)	40.71M	40.80M	+0.21%
`FCN3`	state	internal multi-scale noise	17.84M	17.85M	+0.03%

Table 7 summarizes the model backbones, their prediction targets and noise-entry points, and the parameter counts. Switching from CRPS to DDM only expands the input layers to accept the corrupted target X_t and adds time-conditioning modules for t.

Training Configuration The main pilot experiments use the optimization settings in Table 8. Each training step draws $M = 2$ ensemble members and scores them with the fair CRPS estimator. Evaluation uses EMA weights. Each run targets a single forecast lead (**`12h`**, **`1d`**, **`3d`**, **`5d`**), taking its two input frames separated by the lead (FCN3 uses a single frame, following its native configuration). Widths and depths are set per backbone to a comparable scale and are held fixed across the objectives.

Table 8: Optimization settings, shared by every $2.8125°$ pilot run.

Setting	Value
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Epochs	50
Batch size	16
Gradient clipping	1.0
EMA decay	0.999
Members (M)	2

C.5 Experimental Details at 1.5 degrees

Architectures At $1.5°$, we use the official implementations of MOSAIC (Zhdanov et al., 2026) and U-Cast (Cachay et al., 2026) to compare CRPS training with DDM. Table 9 summarizes the

prediction targets, ensemble-generation mechanisms, and parameter counts. DDM retains each backbone's original prediction parameterization and ensemble-generation mechanism, while additionally conditioning on the corrupted prediction target and its noise level. This conditioning increases the parameter count by 3.23% for MOSAIC and 1.06% for U-Cast.

Table 9: Backbones used in the 1.5° experiments.

			# Parameters		
Model	Target	Ensemble randomness	**CRPS**	**DDM**	**Δ**
MOSAIC	state	latent vector (d=32)	214.19M	221.11M	+3.23%
U-Cast	residual	Monte Carlo dropout	895.43M	904.96M	+1.06%

Training Configuration Under our common experimental setup, we follow the published training recipes of MOSAIC and U-Cast, using only the first training stage for MOSAIC and omitting deep ensembling for U-Cast. The training settings summarized in Table 10 are matched between the CRPS and DDM variants of each backbone. Based on the pilot results in Section D.3.2, both DDM variants use spherical Gaussian corruption noise (SpG) with the heat-kernel schedule ($\kappa_0 = 0.01$).

D Additional Results

D.1 Training Dynamics of Context-Only CRPS Training

We extend the analysis in Section 3.1 by examining training and validation trajectories across all four pilot architectures. As shown in Figure 10, validation nCRPS decreases and gradually levels off at the 12-hour and 1-day horizons. At the 3- and 5-day horizons, it reaches an early minimum and then increases while training loss continues to decrease. This increase is larger at 5 days than at 3 days across all four architectures, indicating more pronounced overfitting at longer forecast horizons.

Training loss uses residual normalization for FGN, Swin, and U-Net and state normalization for FCN3, consistent with the prediction targets in Table 7. Validation nCRPS uses state normalization for all models. The two losses are therefore on different scales for all models except FCN3, and we assess overfitting from their trends during training. Training configurations are provided in Section C.4.

D.2 Proof-of-Concept Experiment in Section 3.1

We provide additional results for the proof-of-concept experiment in Section 3.1, including complete test metrics for FCN3 and an extended experiment with FGN. Both models perform direct 3-day-ahead forecasting at 2.8125° resolution. We compare the CRPS baseline with DDM trained using $t \sim \mathcal{U}[t_{\min}, 1]$ for $t_{\min} \in \{0, 0.25, 0.50, 0.75\}$, together with endpoint-only training at exactly $t = 1$. We train FCN3 for 50 epochs and extend FGN training from 50 to 100 epochs to better observe later training behavior. For each run, we select the checkpoint with the lowest validation nCRPS, evaluated using 240 initializations from 2018–2019 with $M = 50$ ensemble members.

Tables 11 and 12 report test nCRPS and ensemble-mean nRMSE, with percentage changes relative to the corresponding CRPS baseline. Endpoint-only training ($t_{\min} = 1$) closely matches CRPS training, reducing nCRPS by 0.47% for FCN3 and 0.41% for FGN. As discussed in Section B.2, endpoint-only and standard CRPS training share the same population-optimal forecast marginals. Although the learned predictive distribution need not be independent of corruption noise, the similar performance observed here is consistent with this theoretical relationship. Lowering $t_{\min}$ progressively improves both metrics for both architectures. With the full auxiliary conditional denoising tasks ($t_{\min} = 0$), nCRPS and nRMSE decrease by 8.10% and 6.55% for FCN3, and by 9.96% and 8.55% for FGN.

The validation trajectories in Figures 4 and 11 provide a complementary view. For CRPS and endpoint-only training, validation nCRPS increases after an initial improvement, while SSR declines substantially below 1. Broader training noise ranges mitigate this degradation and maintain SSR closer to 1, although underdispersion remains. This qualitative pattern is observed for both FCN3 and FGN. Together, these results are consistent with a regularizing effect of auxiliary conditional

Table 10: Training configurations at 1.5°.

	MOSAIC	U-Cast
Training budget	250,000 steps	100 pretraining / 8 fine-tuning epochs
Effective batch size	16	48
Optimizer	Muon	Muon / AdamW
Peak learning rate	10^{-3}	7×10^{-3} / 7×10^{-5}
Learning-rate schedule	cosine	linear warmup + cosine
Final learning rate	10^{-6}	10^{-8}
Warmup steps	0	1,500
Weight decay	10^{-2}	0 / 0.1
Training ensemble size	2	2
Gradient-norm clipping	1.0	1.0

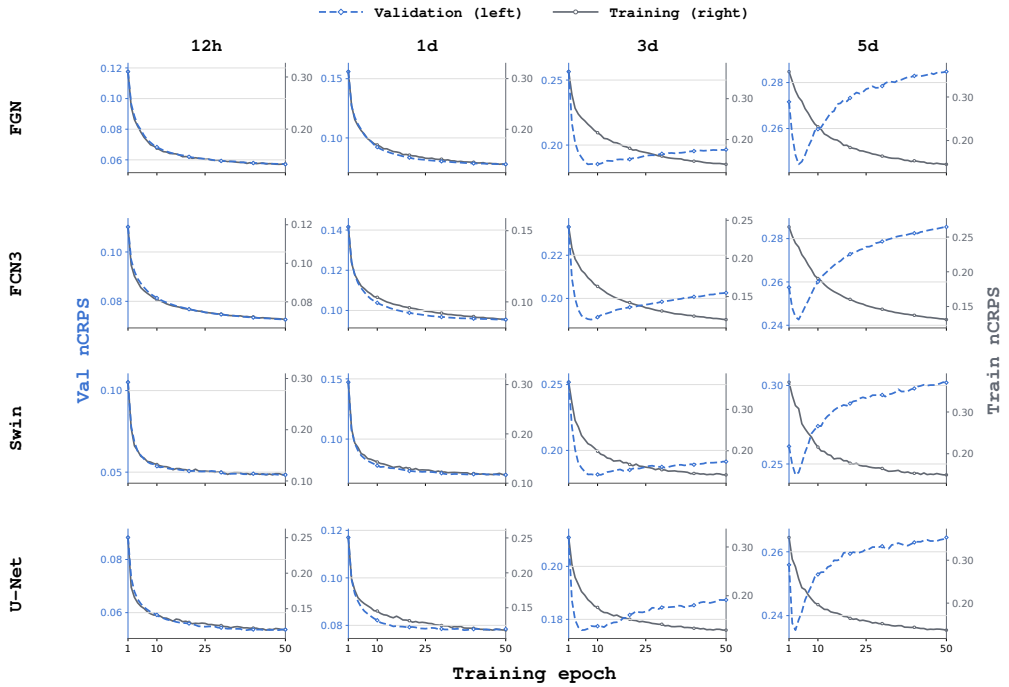


Figure 10: Training loss (solid, right axis) and validation nCRPS (dashed, left axis) of CRPS-trained FGN, FCN3, Swin, and U-Net across forecast horizons at 2.8125°. Training loss uses residual normalization for FGN, Swin, and U-Net and state normalization for FCN3, whereas validation nCRPS uses state normalization for all models. Training and validation losses therefore share the same normalization only for FCN3. Training configurations are provided in Section C.4.

denoising tasks learned through shared parameters, supporting the hypothesis that these tasks can improve one-step forecasting and provide better generalization.

D.3 Additional Results for the 2.8125° Pilot Experiments

D.3.1 Direct Forecasting and Autoregressive Rollout Performance

Table 13 reports one-step forecasting performance for all four backbones, and Tables 14 and 15 extend the rollout evaluation of FGN and FCN3. As in Tables 1 and 2, DDM generally improves nCRPS and nRMSE, with larger gains at longer rollout intervals that partly persist during rollouts.

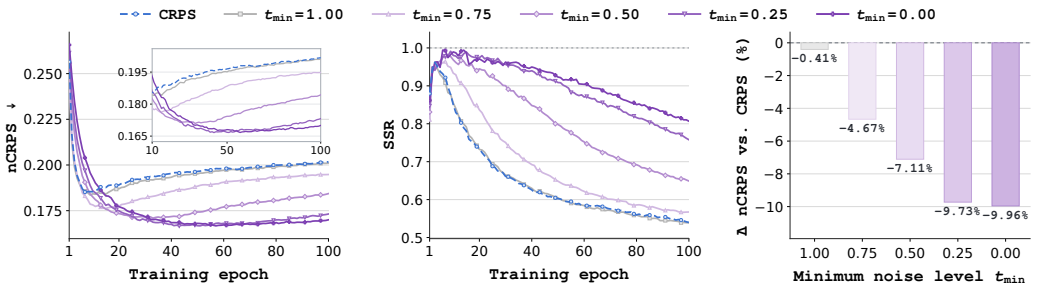


Figure 11: **Effect of conditional denoising tasks on FGN one-step forecasting.** FGN is trained for 100 epochs for 3-day-ahead forecasting at 2.8125°. Panels show validation nCRPS (left), validation SSR (middle), and test nCRPS changes (%) relative to the CRPS baseline (right). Test results use checkpoints selected by validation nCRPS, with NFE $= 1$ for all DDM configurations.

Table 11: Effect of $t_{\min}$ on FCN3-`3d` one-step forecasting performance at 2.8125° used in Figure 4.

Objective	Training t	**nCRPS** ↓	**nRMSE** ↓
CRPS	–	0.1915	0.3920
DDM	$t = 1$	0.1906 (−0.47%)	0.3906 (−0.36%)
	$t \sim \mathcal{U}[0.75, 1]$	0.1853 (−3.21%)	0.3822 (−2.52%)
	$t \sim \mathcal{U}[0.50, 1]$	0.1809 (−5.53%)	0.3749 (−4.37%)
	$t \sim \mathcal{U}[0.25, 1]$	0.1780 (−7.03%)	0.3699 (−5.65%)
	$t \sim \mathcal{U}[0, 1]$	0.1759 (−8.10%)	0.3664 (−6.55%)

D.3.2 Noise Prior and Schedule Ablation

To assess the sensitivity of one-step forecasting to the corruption process, we compare spatial noise priors and corruption schedules while keeping the remaining DDM settings fixed (Table 16). Here, x is the normalized prediction target and ε is corruption noise, distinct from the randomness source ζ.

Standard diffusion models typically use independent Gaussian noise at each grid point (`Grid`) (Ho et al., 2020). GenCast reports small but consistent improvements from sampling isotropic Gaussian noise in the spherical-harmonic domain (Price et al., 2025). Motivated by this finding, we evaluate spherical Gaussian (`SpG`) noise with equal variance across retained harmonic modes, generated as

$$\varepsilon_{\mathrm{SpG}}(\omega) = g \sum_{\ell=0}^{L_{\mathrm{noise}}} \sum_{m=-\ell}^{\ell} a_{\ell m}\, \psi_{\ell m}(\omega), \qquad a_{\ell m} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \tag{29}$$

where $\omega \in \mathbb{S}^2$, $\psi_{\ell m}$ are real orthonormal spherical harmonics, L_{noise} is the maximum degree, and g is a normalization constant. We sample this isotropic Gaussian field independently across channels.

Linear interpolation attenuates all target coefficients by the same factor. We therefore consider a spherical heat-kernel (`HK`) schedule, inspired by Hoogeboom & Salimans (2023), that attenuates fine-scale target information more rapidly.

Since spherical harmonics of degree ℓ have Laplace–Beltrami eigenvalue $-\ell(\ell+1)$ on the unit sphere (Atkinson & Han, 2012), we define the interpolation for degrees $0 \le \ell \le L_{\mathrm{HK}}$ as

$$t_\ell(t) = 1 - (1-t)\exp\left[-\frac{\kappa_0 t}{2}\ell(\ell+1)\right], \qquad (x_t)_{\ell m} = \big[1 - t_\ell(t)\big] x_{\ell m} + t_\ell(t)\varepsilon_{\ell m}. \tag{30}$$

Here, $x_{\ell m}$ and $\varepsilon_{\ell m}$ are the spherical-harmonic coefficients of the target and noise, L_{HK} is the maximum degree, and $\kappa_0 \ge 0$

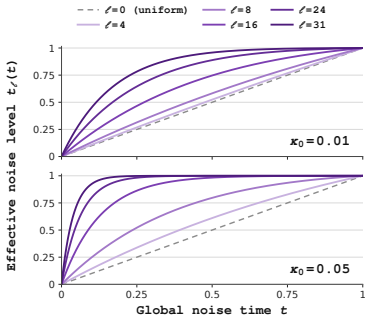


Figure 12: Effective noise levels by harmonic degree for $\kappa_0 = 0.01$ (top) and $\kappa_0 = 0.05$ (bottom).

Table 12: Effect of $t_{\min}$ on FGN-3d-100ep one-step forecasting at 2.8125° used in Figure 11. Note that the best checkpoint of the CRPS baseline is the same as FGN-3d-50ep due to overfitting.

Objective	Training t	nCRPS ↓	nRMSE ↓
CRPS	–	0.1868	0.3847
DDM	$t = 1$	0.1861 (−0.41%)	0.3828 (−0.49%)
	$t \sim \mathcal{U}[0.75, 1]$	0.1781 (−4.67%)	0.3685 (−4.20%)
	$t \sim \mathcal{U}[0.50, 1]$	0.1736 (−7.11%)	0.3611 (−6.13%)
	$t \sim \mathcal{U}[0.25, 1]$	0.1687 (−9.73%)	0.3528 (−8.27%)
	$t \sim \mathcal{U}[0, 1]$	0.1682 (−9.96%)	0.3517 (−8.55%)

Table 13: Direct forecasting performance at 2.8125° across four architectures.

		nCRPS ↓				nRMSE ↓			
Model	Objective	12h	1d	3d	5d	12h	1d	3d	5d
Swin	CRPS	0.0491	0.0712	0.1839	0.2498	0.0966	0.1429	0.3789	0.5151
	DDM	0.0480	0.0698	0.1628	0.2264	0.0947	0.1406	0.3409	0.4739
	Δ (%)	-2.1	-2.0	-11.5	-9.4	-1.9	-1.6	-10.0	-8.0
U-Net	CRPS	0.0547	0.0799	0.1759	0.2333	0.1072	0.1581	0.3608	0.4782
	DDM	0.0543	0.0789	0.1687	0.2226	0.1064	0.1579	0.3507	0.4632
	Δ (%)	-0.7	-1.3	-4.1	-4.6	-0.7	-0.1	-2.8	-3.1
FGN	CRPS	0.0581	0.0792	0.1868	0.2463	0.1138	0.1586	0.3847	0.5034
	DDM	0.0577	0.0778	0.1692	0.2290	0.1130	0.1559	0.3542	0.4760
	Δ (%)	-0.8	-1.8	-9.4	-7.0	-0.7	-1.7	-7.9	-5.4
FCN3	CRPS	0.0736	0.0967	0.1915	0.2452	0.1461	0.1937	0.3920	0.5029
	DDM	0.0735	0.0942	0.1759	0.2283	0.1461	0.1894	0.3664	0.4755
	Δ (%)	-0.1	-2.6	-8.1	-6.9	-0.0	-2.2	-6.6	-5.5

controls the attenuation strength. As shown in Figure 12, higher degrees have larger $t_\ell(t)$ and smaller target weights $1 - t_\ell(t)$ at the same t. Larger κ_0 strengthens this scale dependence.

In practice, g is computed analytically to ensure unit grid-averaged marginal variance. We use $(L_{\text{noise}}, L_{\text{HK}}) = (63, 31)$ at 2.8125° and $(120, 60)$ at 1.5°. The residual outside the retained representation is preserved and interpolated using $t_{L_{\text{HK}}}(t)$. The `Grid` and `SpG` variants use linear interpolation, with independent standard Gaussian grid noise and spherical Gaussian noise, respectively. The `HK` variants use the same spherical Gaussian prior with $\kappa_0 \in \{0.01, 0.05\}$.

Table 16 shows that `HK` generally achieves the lowest nCRPS at 12h and 1d, whereas `Grid` or `SpG` performs best at 3d and 5d. Thus, the preferred noise distribution and schedule depend on the rollout interval and backbone. Nevertheless, all DDM configurations outperform the CRPS baseline at the two longer intervals across all four backbones. For the main pilot experiments, we select the noise prior and schedule with the lowest validation nCRPS for each backbone and forecast horizon.

D.4 Additional Results at 1.5° High-dimensional Global Weather Forecasting

D.4.1 Forecast Performance

We provide channel-wise comparisons and rollout curves using all 732 initializations at 00 and 12 UTC in `2020`, with $M = 50$ ensemble members and NFE $= 1$ for DDM. Figure 13 compares DDM and CRPS training across all 84 forecast channels over 10-day rollouts. In addition to IFS-ENS, Table 3 reports results for GenCast (Price et al., 2025) and NeuralGCM (Kochkov et al., 2024). These

Table 14: Autoregressive rollout nCRPS at 2.8125°.

Model	RI	Objective	24h	72h	120h	240h	288h	360h
FGN	12h	CRPS	0.0747	0.1420	0.1928	0.2442	0.2514	0.2571
		DDM	0.0743	0.1419	0.1926	0.2443	0.2514	0.2570
		Δ (%)	-0.5	-0.1	-0.1	+0.0	+0.0	-0.0
	1d	CRPS	0.0792	0.1434	0.1934	0.2444	0.2509	0.2566
		DDM	0.0778	0.1412	0.1910	0.2428	0.2499	0.2559
		Δ (%)	-1.8	-1.6	-1.2	-0.7	-0.4	-0.3
	3d	CRPS	–	0.1868	–	–	0.2628	0.2659
		DDM	–	0.1692	–	–	0.2538	0.2582
		Δ (%)	–	-9.4	–	–	-3.4	-2.9
	5d	CRPS	–	–	0.2463	0.2652	–	0.2701
		DDM	–	–	0.2290	0.2557	–	0.2621
		Δ (%)	–	–	-7.0	-3.6	–	-3.0
FCN3	12h	CRPS	0.0922	0.1634	0.2105	0.2512	0.2560	0.2602
		DDM	0.0923	0.1632	0.2100	0.2507	0.2556	0.2600
		Δ (%)	+0.1	-0.1	-0.2	-0.2	-0.2	-0.1
	1d	CRPS	0.0967	0.1641	0.2107	0.2513	0.2561	0.2608
		DDM	0.0942	0.1597	0.2058	0.2487	0.2541	0.2588
		Δ (%)	-2.6	-2.7	-2.3	-1.0	-0.8	-0.8
	3d	CRPS	–	0.1915	–	–	0.2620	0.2656
		DDM	–	0.1759	–	–	0.2544	0.2585
		Δ (%)	–	-8.1	–	–	-2.9	-2.7
	5d	CRPS	–	–	0.2452	0.2612	–	0.2662
		DDM	–	–	0.2283	0.2527	–	0.2593
		Δ (%)	–	–	-6.9	-3.2	–	-2.6

baselines are included for reference, as their configurations, including spatial resolution, differ from ours. Our primary focus is comparing DDM and CRPS training under matched experimental settings.

Comparisons use native forecast intervals of 24 h for MOSAIC and 12 h for U-Cast, yielding 840 and 1,680 channel–lead pairs, respectively. DDM achieves lower CRPS in 93.7% (787/840) of these pairs for MOSAIC and 82.0% (1,377/1,680) for U-Cast. The corresponding fractions for ensemble-mean RMSE are 92.0% (773/840) and 81.0% (1,360/1,680). Improvements extend across surface variables and most tropospheric levels, whereas degradations are concentrated at the lowest-pressure levels.

We next visualize autoregressive rollout curves on 19 forecast channels common to our models and the publicly available WeatherBench 2 IFS-ENS archive (Rasp et al., 2024) (Figures 14–16). For both backbones, DDM achieves lower CRPS than the corresponding CRPS-trained model at every evaluated channel–lead pair in this comparison, although several U-Cast differences fall below the precision displayed in Table 3. Ensemble-mean RMSE improves across all pairs for MOSAIC and nearly all pairs for U-Cast. The SSR curves show that DDM generally moves the SSR closer to 1.

We additionally evaluate MOSAIC with NFE $= 2$ under the same evaluation protocol. As shown in Figure 17, two-step sampling retains lower CRPS in all 787 channel–lead pairs where one-step sampling outperforms CRPS-trained MOSAIC and achieves lower CRPS in 16 additional pairs,

Table 15: Autoregressive rollout nRMSE at 2.8125°. The same runs and conventions as Table 14.

Model	RI	Objective	24h	72h	120h	240h	288h	360h
FGN	12h	CRPS	0.1497	0.2986	0.4060	0.5082	0.5220	0.5332
		DDM	0.1490	0.2985	0.4056	0.5085	0.5223	0.5329
		Δ (%)	-0.4	-0.0	-0.1	+0.1	+0.1	-0.0
	1d	CRPS	0.1586	0.3012	0.4066	0.5083	0.5210	0.5321
		DDM	0.1559	0.2978	0.4031	0.5062	0.5199	0.5314
		Δ (%)	-1.7	-1.1	-0.9	-0.4	-0.2	-0.1
	3d	CRPS	–	0.3847	–	–	0.5380	0.5441
		DDM	–	0.3542	–	–	0.5258	0.5345
		Δ (%)	–	-7.9	–	–	-2.3	-1.8
	5d	CRPS	–	–	0.5034	0.5408	–	0.5501
		DDM	–	–	0.4760	0.5277	–	0.5402
		Δ (%)	–	–	-5.4	-2.4	–	-1.8
FCN3	12h	CRPS	0.1853	0.3399	0.4391	0.5209	0.5303	0.5384
		DDM	0.1855	0.3394	0.4378	0.5197	0.5293	0.5379
		Δ (%)	+0.1	-0.1	-0.3	-0.2	-0.2	-0.1
	1d	CRPS	0.1937	0.3401	0.4381	0.5207	0.5301	0.5394
		DDM	0.1894	0.3327	0.4297	0.5164	0.5270	0.5361
		Δ (%)	-2.2	-2.2	-1.9	-0.8	-0.6	-0.6
	3d	CRPS	–	0.3920	–	–	0.5368	0.5437
		DDM	–	0.3664	–	–	0.5273	0.5354
		Δ (%)	–	-6.6	–	–	-1.8	-1.5
	5d	CRPS	–	–	0.5029	0.5357	–	0.5445
		DDM	–	–	0.4755	0.5243	–	0.5373
		Δ (%)	–	–	-5.5	-2.1	–	-1.3

bringing the total to 803 out of 840. These results illustrate the potential benefits of refinement without retraining, at the cost of one additional model evaluation per member and rollout step.

D.4.2 Robustness to Distribution Shift

In the distribution shift experiment, we initialize and verify forecasts with HRES-fc0. As in the ERA5 evaluation, we use all 732 initializations at 00 and 12 UTC in 2020, with $M = 50$ ensemble members and NFE $= 1$ for DDM. Note that the SSR deviation reported in Table 4 refers to $|\mathrm{SSR} - 1|$ averaged over all channels and lead times, with smaller values indicating better spread–skill balance.

To compare with the IFS-ENS baseline, Figures 18–20 show rollout curves for CRPS, ensemble-mean RMSE and SSR for the 19 channels shared with IFS-ENS. DDM achieves lower CRPS and ensemble-mean RMSE than the CRPS-trained model across all 190 channel–lead pairs and the spread–skill ratio is also generally closer to 1, indicating better agreement between ensemble spread and forecast error. These results show that the advantages of DDM persist under the ERA5–HRES shift.

Table 16: Noise scheduling for each backbone and RI at $2.8125°$, with the single-step nCRPS. Values are test nCRPS. Shaded cells denote the configuration selected by the lowest validation nCRPS.

		CRPS	DDM				
Model	RI	baseline	grid	SpG	HK κ_0=0.01	HK κ_0=0.05	Δ(%)
Swin	12h	0.0491	0.0489	0.0496	0.0484	0.0480	-2.1
	1d	0.0712	0.0712	0.0708	0.0698	0.0704	-2.0
	3d	0.1839	0.1644	0.1628	0.1645	0.1705	-11.5
	5d	0.2498	0.2264	0.2268	0.2297	0.2358	-9.4
U-Net	12h	0.0547	0.0553	0.0561	0.0548	0.0543	-0.7
	1d	0.0799	0.0789	0.0797	0.0800	0.0801	-1.3
	3d	0.1759	0.1687	0.1692	0.1692	0.1718	-4.1
	5d	0.2333	0.2226	0.2237	0.2268	0.2280	-4.6
FGN	12h	0.0581	0.0592	0.0592	0.0577	0.0578	-0.8
	1d	0.0792	0.0797	0.0793	0.0778	0.0782	-1.8
	3d	0.1868	0.1697	0.1692	0.1704	0.1775	-9.4
	5d	0.2463	0.2298	0.2290	0.2303	0.2356	-7.0
FCN3	12h	0.0736	0.0743	0.0740	0.0735	0.0736	-0.1
	1d	0.0967	0.0956	0.0950	0.0942	0.0945	-2.6
	3d	0.1915	0.1761	0.1759	0.1784	0.1845	-8.1
	5d	0.2452	0.2283	0.2295	0.2314	0.2372	-6.9

D.4.3 Tropical Cyclone Tracking

We provide additional tropical-cyclone tracking results for Hurricanes Ian and Michael using 50-member ensembles from MOSAIC and U-Cast, with IBTrACS best tracks as the reference. Figure 21 shows the ensemble track forecasts for Hurricane Michael, and Table 17 reports ensemble-mean track error and track CRPS for both storms. DDM improves both metrics over CRPS training.

To examine the ensemble forecast positions beyond the ensemble mean, Figures 22–24 show enlarged views of Hurricane Ian forecasts with circles of radius 150, 200, and 300 km around the time-matched IBTrACS center. The highlighted lead times are 48 h for MOSAIC and 120 h for U-Cast. Table 18 summarizes the number and percentage of members within each radius, using all 50 members regardless of whether individual positions fall inside the displayed crop. Across these radii, DDM places more members near the reference center than CRPS training for both backbones. At 200 km, the fraction increases from 62% to 94% for MOSAIC and from 36% to 46% for U-Cast.

Tracking and Metrics We track storm centers using MSLP minima following Zhdanov et al. (2026). Each member track is initialized at the IBTrACS (Knapp et al., 2010) position and subsequently follows the minimum MSLP within a search radius around its previous center. Forecasts are evaluated against the time-matched IBTrACS best track during the tropical phase.

At each lead time, let $\{\mathbf{x}_m\}_{m=1}^{M}$ denote the predicted storm positions and $\mathbf{y}$ the observed position. We measure great-circle distance d using the haversine formula with Earth radius $R = 6371$ km. The ensemble-mean track error is $d(\bar{\mathbf{x}}, \mathbf{y})$, where $\bar{\mathbf{x}}$ is the arithmetic mean of the member latitude–longitude coordinates. Following TCBench (Gomez et al., 2026), we compute track CRPS as

$$\text{TrackCRPS} = \frac{1}{M}\sum_{m=1}^{M} d(\mathbf{x}_m, \mathbf{y}) - \frac{1}{2M(M-1)}\sum_{i=1}^{M}\sum_{\substack{j=1 \\ j\neq i}}^{M} d(\mathbf{x}_i, \mathbf{x}_j). \tag{31}$$

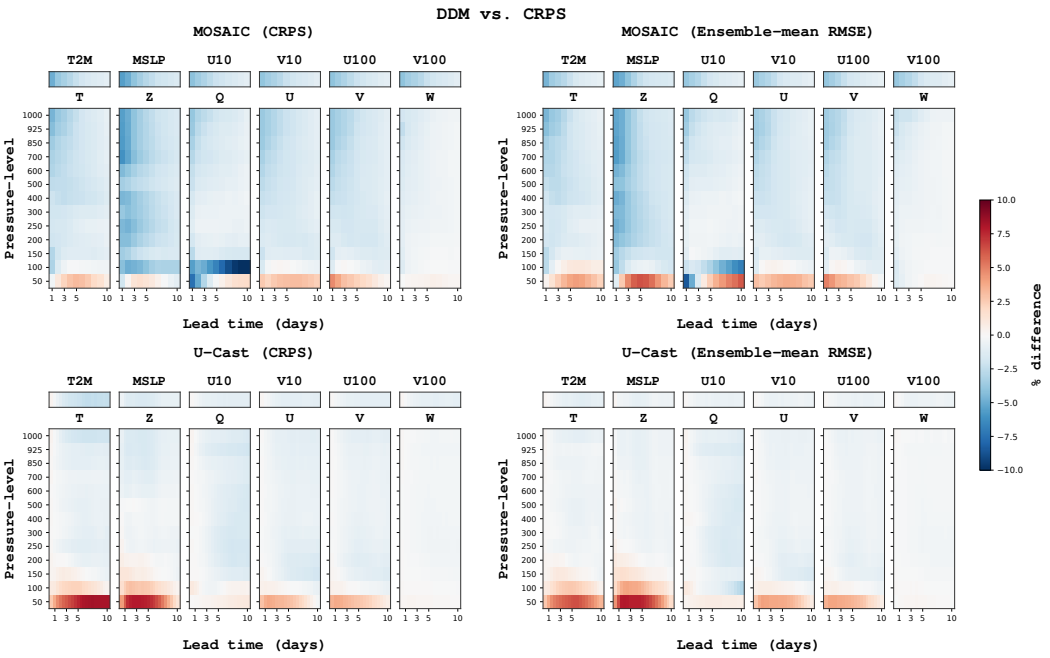


Figure 13: Relative performance differences in CRPS (left) and ensemble-mean RMSE (right) between DDM and CRPS training for MOSAIC (top) and U-Cast (bottom). Negative values (blue) indicate that DDM shows better performance than CRPS.

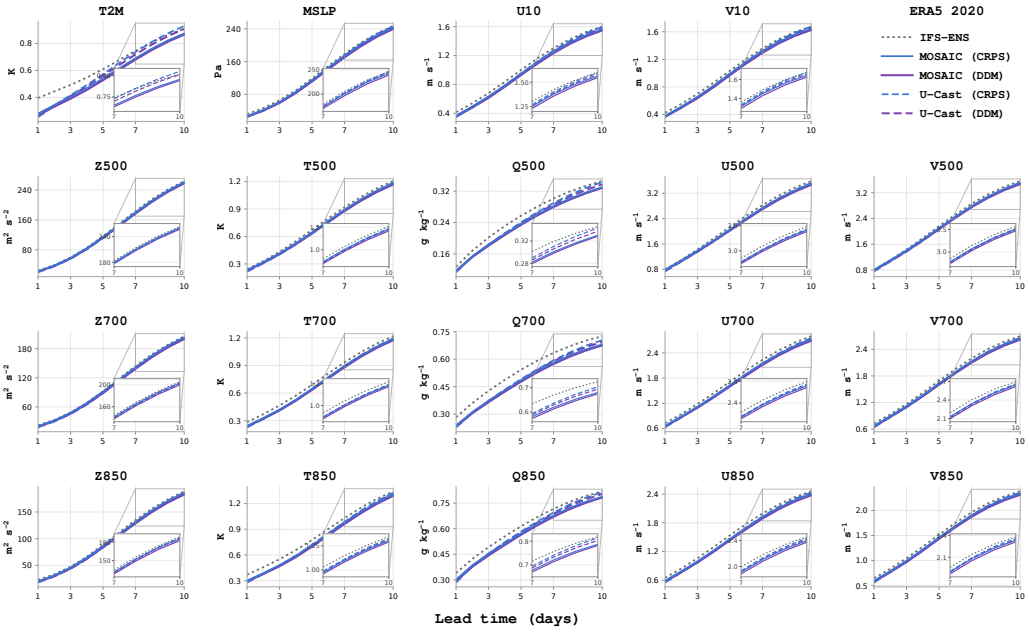


Figure 14: CRPS over 10-day autoregressive rollouts in ERA5 2020.

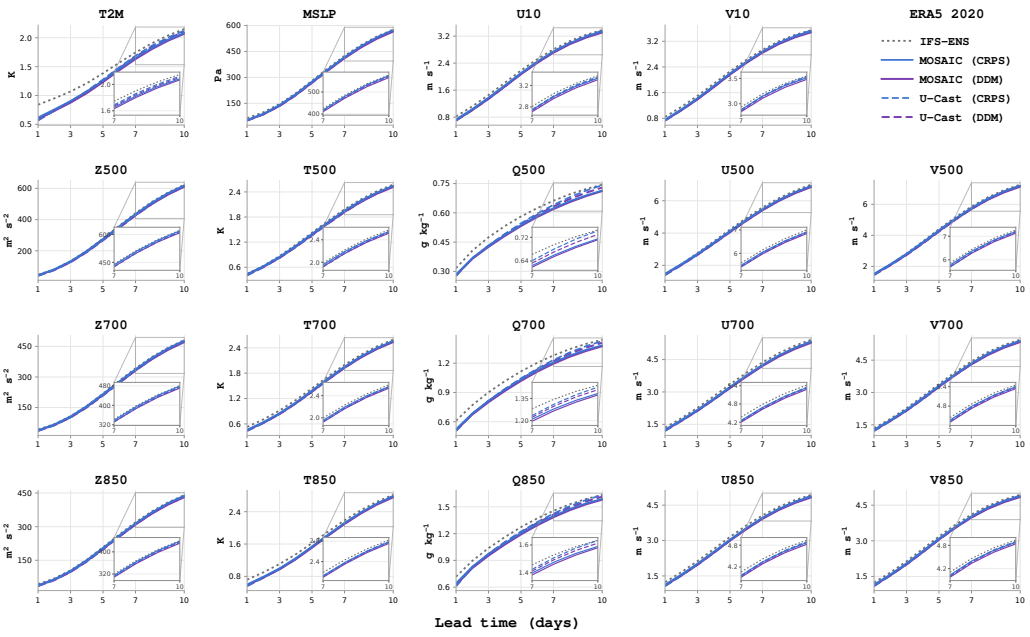


Figure 15: Ensemble-mean RMSE over 10-day autoregressive rollouts in ERA5 2020.

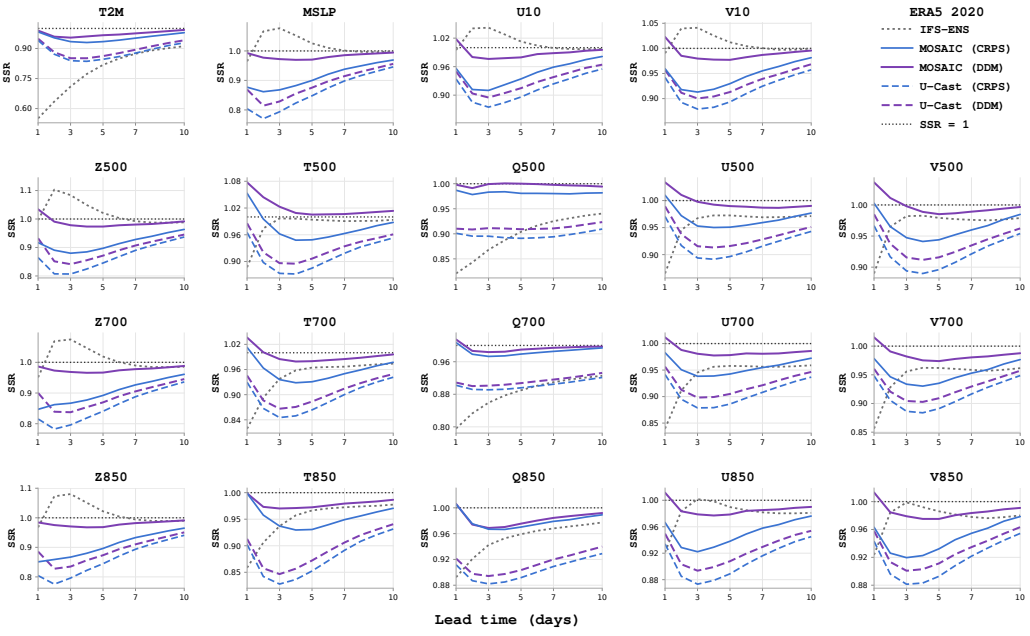


Figure 16: SSR over 10-day autoregressive rollouts in ERA5 2020.

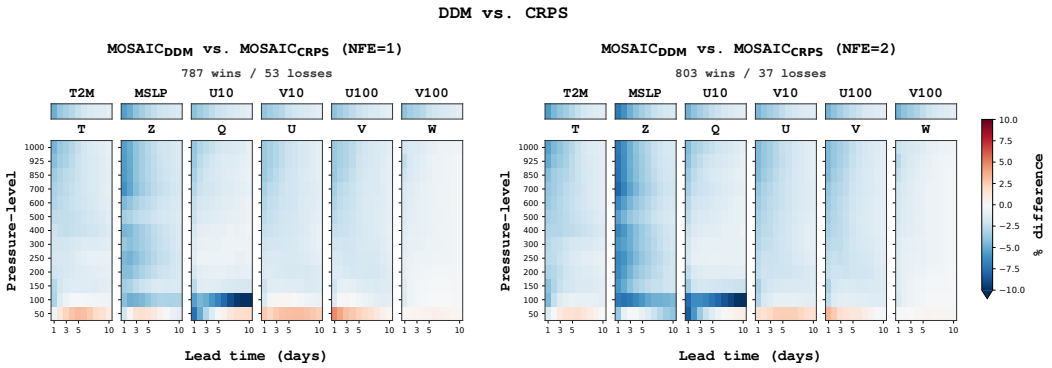


Figure 17: Effect of NFE on MOSAIC forecasting performance. Relative CRPS differences between DDM and CRPS-trained MOSAIC across 84 forecast channels over 10-day autoregressive rollouts. The same DDM checkpoint is evaluated with NFE $= 1$ (left) and NFE $= 2$ (right) in ERA5 2020.

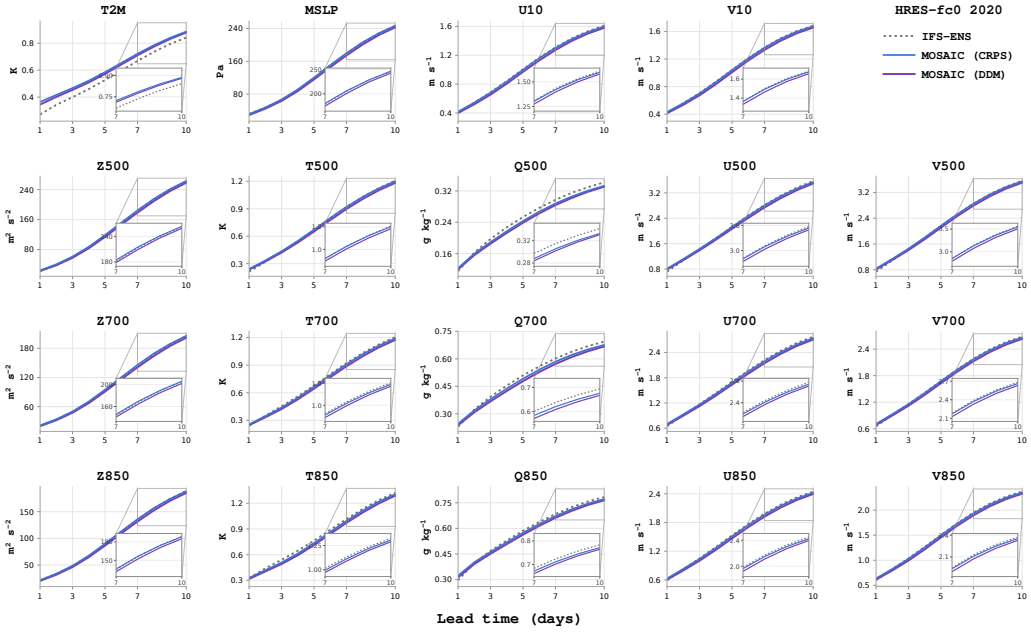


Figure 18: CRPS over 10-day autoregressive rollouts in HRES-fc0 2020.

Table 17: Tropical-cyclone ensemble-mean track error and track CRPS (km).

		Hurricane Ian						Hurricane Michael							
		Track Error ↓			Track CRPS ↓			Track Error ↓				Track CRPS ↓			
Method	Objective	1d	3d	5d	1d	3d	5d	1d	2d	3d	4d	1d	2d	3d	4d
MOSAIC	**CRPS**	73	89	189	51	68	125	42	81	156	231	30	54	110	153
MOSAIC	**DDM**	38	44	55	26	39	63	25	17	68	138	17	17	47	86
U-Cast	**CRPS**	39	59	245	30	43	148	36	75	150	308	25	51	112	231
U-Cast	**DDM**	35	37	210	24	38	123	35	62	107	210	24	40	72	141

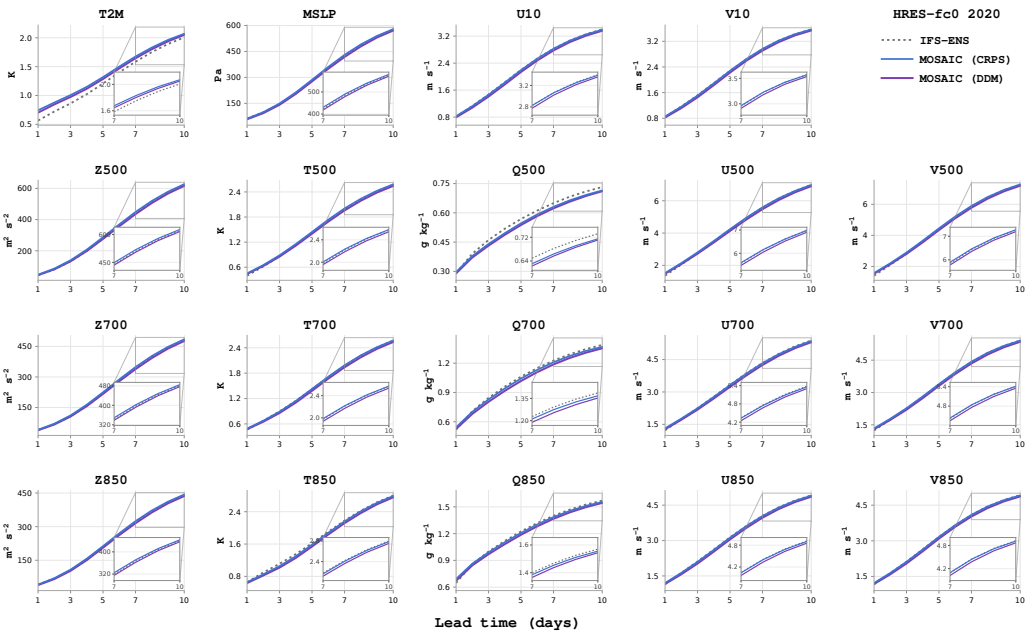


Figure 19: Ensemble-mean RMSE over 10-day autoregressive rollouts in HRES-fc0 2020.

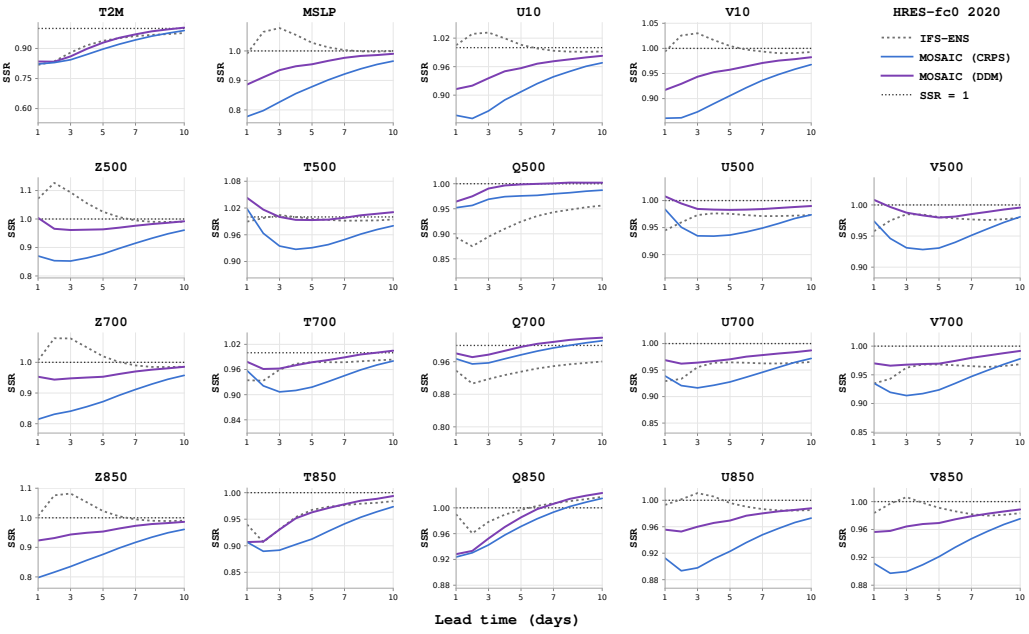


Figure 20: SSR over 10-day autoregressive rollouts in HRES-fc0 2020.

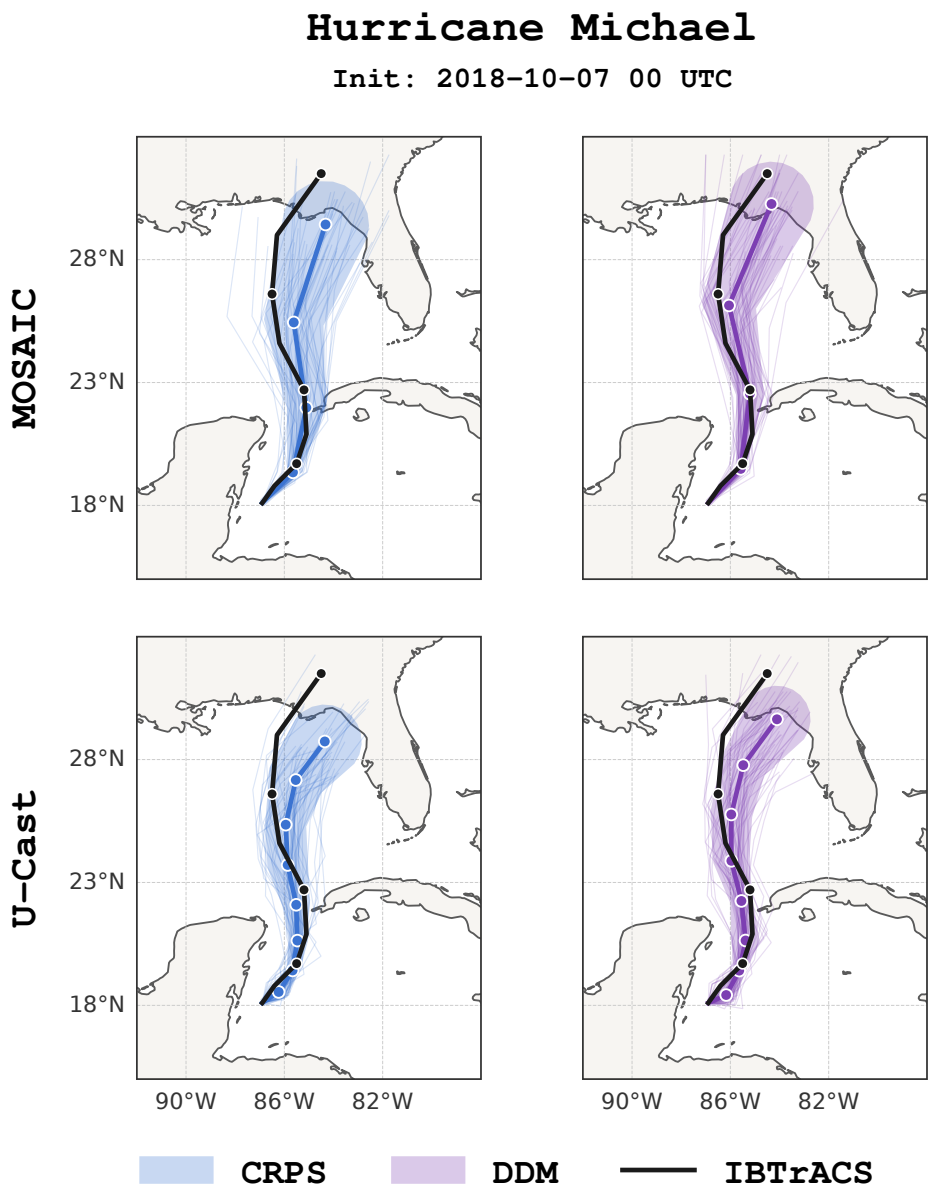


Figure 21: Hurricane Michael track forecasts from 50-member MOSAIC (top) and U-Cast (bottom) ensembles, comparing CRPS (blue, left) and DDM (purple, right). Michael occurred in 2018, within the validation period, and is included as an illustrative case study.

Table 18: Number and percentage of ensemble members within 150, 200, and 300 km of the time-matched IBTrACS center for Hurricane Ian. MOSAIC is evaluated at 48 h and U-Cast at 120 h, corresponding to the inset lead times as shown in Figures 22–24.

Method	Objective	Lead time	150 km	200 km	300 km
MOSAIC	**CRPS**	48 h	11/50 (22%)	31/50 (62%)	48/50 (96%)
MOSAIC	**DDM**	48 h	23/50 (46%)	47/50 (94%)	50/50 (100%)
U-Cast	**CRPS**	120 h	13/50 (26%)	18/50 (36%)	28/50 (56%)
U-Cast	**DDM**	120 h	16/50 (32%)	23/50 (46%)	31/50 (62%)

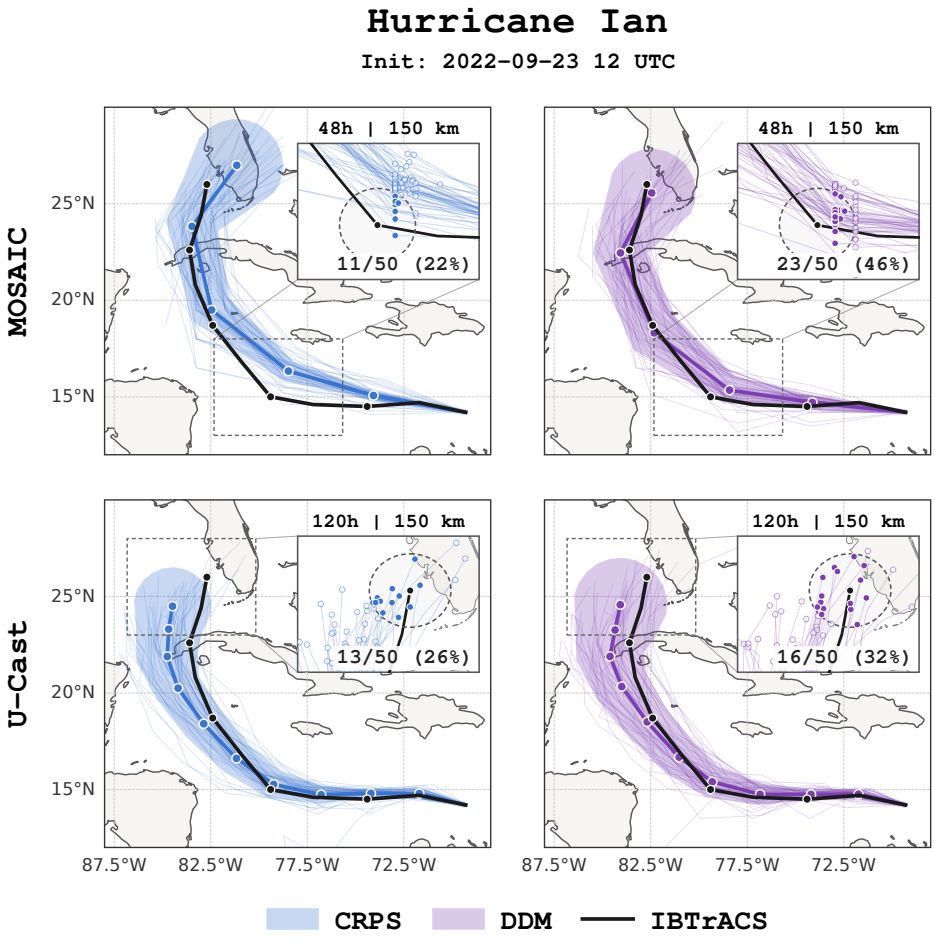


Figure 22: 50-member ensemble track forecasts for Hurricane Ian initialized at 12 UTC on 23 September 2022, from MOSAIC (top) and U-Cast (bottom), comparing CRPS (blue, left) and DDM (purple, right). Insets show the 48 h MOSAIC and 120 h U-Cast forecast positions. Dashed circles mark a 150 km radius around the time-matched IBTrACS center.

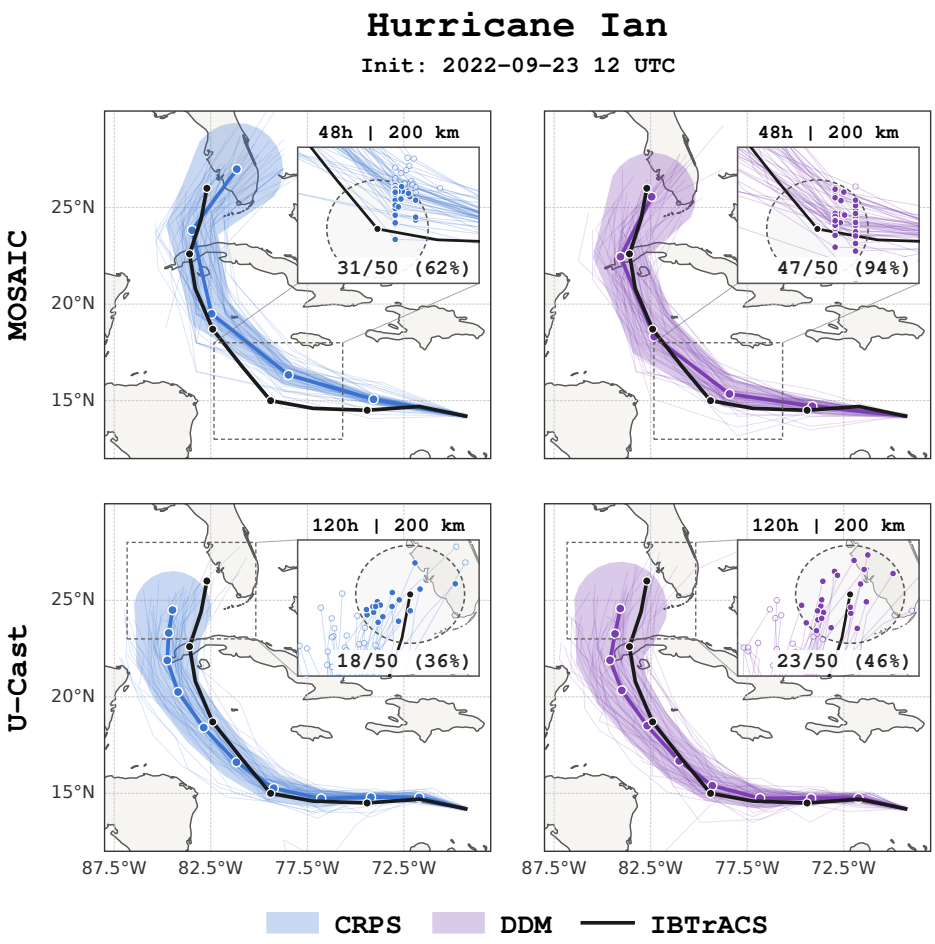


Figure 23: Same as Figure 22, but with a 200 km radius.

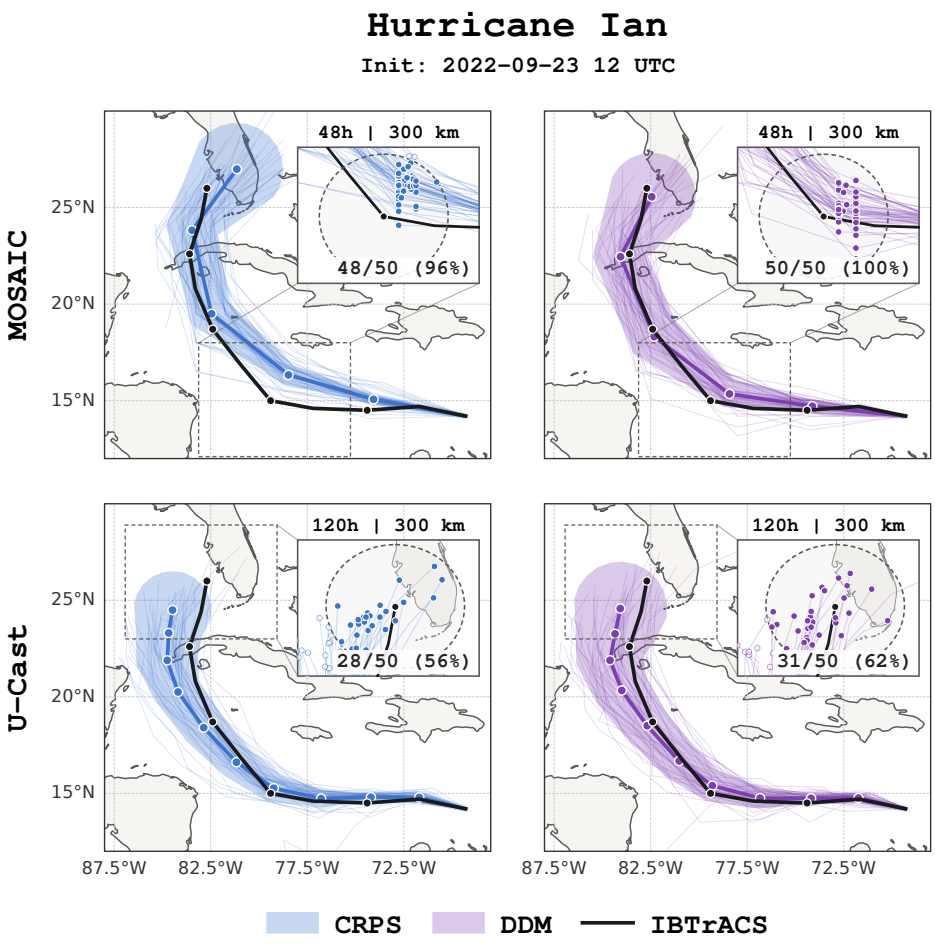


Figure 24: Same as Figure 22, but with a 300 km radius.