XAURORA: GENERATIVE WEATHER FORECASTING WITH DENOISING STOCHASTIC INTERPOLANTS FROM A FOUNDATION MODEL PRIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning has revolutionised weather forecasting in recent years, especially through atmospheric foundation models, which offer competitive skill for a fraction of the computational costs of classic physics-based models. However, most existing foundation models are deterministic, limiting the generation of large ensembles for accurate uncertainty quantification, extreme weather risk assessment, and long-range weather forecasting. Furthermore, these models incur a large, often prohibitive, computational overhead to train from scratch. To address these shortcomings, we turn a pretrained deterministic prior model, namely the Aurora foundation model, into a generative ensemble-prediction model. To that end, we introduce a novel generative method, *Denoising Stochastic Interpolants*, combined with a *replay buffer* for Stochastic Differential Equation (SDE) rollout, enabling probabilistic training of SDE trajectories. Our stochastic foundation model, *Xaurora*, is finetuned from the small Aurora version, yet it approaches the state-of-the-art on global ensemble metrics and is competitive with the large version of Aurora. Our method is parameter and sample efficient, and generates skilful 15-day forecasts in 13 minutes. Our results demonstrate that deterministic foundation models can be efficiently extended into even stronger stochastic models.

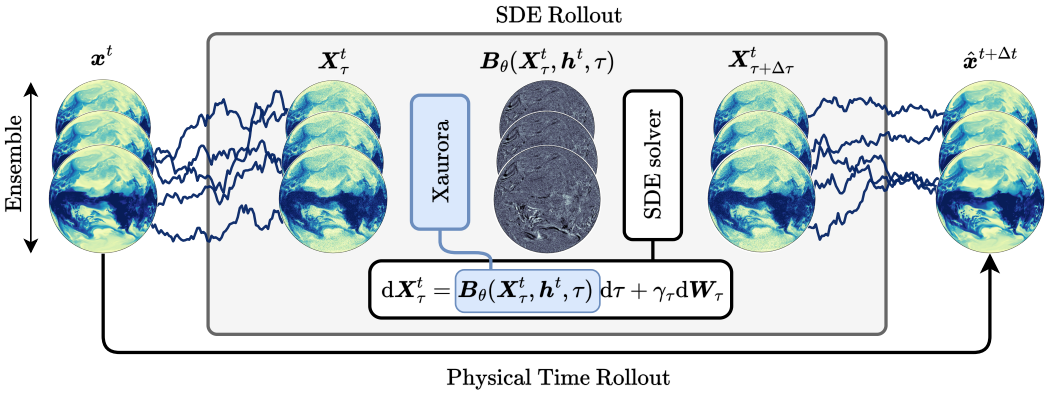

Figure 1: **Xaurora produces ensemble forecasts with a nested rollout**. Each physical step $\boldsymbol{x}^t \to \hat{\boldsymbol{x}}^{t+\Delta t}$ is generated by integrating a stochastic differential equation (SDE) over pseudo-time $\tau \in [0, 1]$ (SDE time, grey box). Starting from the current state $\boldsymbol{X}_0^t = \boldsymbol{x}^t$, the state $\boldsymbol{X}_\tau^t$ follows an SDE formed by the Xaurora drift term ($\boldsymbol{B}_\theta$) and a Wiener process ($\boldsymbol{W}_\tau$). At $\tau = 1$, the terminal SDE state is the forecast, $\hat{\boldsymbol{x}}^{t+\Delta t} := \boldsymbol{X}_1^t$, and the procedure is repeated autoregressively for the physical time rollout. Ensemble members are generated by sampling different Wiener processes $\boldsymbol{W}_\tau$.

1 INTRODUCTION

Accurate weather forecasting is paramount for human activities, from daily life to high-stakes decision making in public safety, agriculture, or transportation. However, the atmosphere is a chaotic system, its predictability is fundamentally limited, with forecast errors growing non-linearly as lead times

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.

increase (Lorenz, 1969; 1982). Therefore, a forecast by itself has limited value without reliable uncertainty quantification, a crucial element for decision making or risk assessment of extreme events. Hence, weather agencies and decision makers rely on ensemble forecasts, which, however, come with substantial computational cost, limiting their size.

Recently, Deep Learning Weather Prediction (DLWP) models matched the performance of traditional Numerical Weather Prediction (NWP) models for only a fraction of the inference cost, offering the community a new avenue for forecasting. However, training ensemble DLWP models is computationally very expensive (Lang et al., 2024b). On the other hand, deterministic foundation models, such as Aurora (Bodnar et al., 2025), achieve remarkable forecasting performance with a modest cost, and can be finetuned for different applications. Instead of training stochastic models from scratch, we propose to build generative models using these pretrained deterministic models as priors. Our strategy is motivated by prior work showing that deep neural networks learn generalisable features that can be transferred to related tasks (Yosinski et al., 2014; Devlin et al., 2019; Raffel et al., 2020) and accommodate new parameters (Houlsby et al., 2019; Hu et al., 2021; Zhang et al., 2023). Capitalising on these concepts, we finetune Aurora into a generative model with only minimal architectural changes. We call our method *Xaurora*, short for *eXtended Aurora*.

We postulate that state-of-the-art generative methods like Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020) and Flow Matching (FM) (Lipman et al., 2023) are ill-suited for our task, since they generate states from pure noise. Instead, we employ Stochastic Interpolants (SI) (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2023; Chen et al., 2024) that build a stochastic bridge between two consecutive states, which is more suited for finetuning autoregressive state prediction models. We further introduce a new parameterisation, Denoising Stochastic Interpolants, enabling a warm initialisation from a pretrained deterministic prior. We also propose an efficient rollout training strategy for SDE-based ensemble generative models. Our contributions are as follows:

1. **Denoising Stochastic Interpolant.** A new SI parameterisation designed for warm starts: it properly exploits Aurora's accuracy at initialisation while remaining stable in very high-dimensional pixel space (Li & He, 2026);
2. **Finetuning strategy.** We generalise Aurora's replay buffer-based rollout training approach to the optimisation of generative models, combining SDE rollout with a single-backpropagation Continuous Ranked Probability Score (CRPS) (Gneiting & Raftery, 2007) objective;
3. **Performance.** Applied to the Aurora-Small model, the method reaches ensemble mean unbiased RMSE and CRPS close to state-of-the-art (SOTA) operational NWP/DLWP systems, and outperforms all non-operational Aurora models on most variables and lead times despite using $10\times$ fewer parameters. Furthermore, we perform a spectral calibration analysis showing that variable-dependent, spatially-correlated, isotropic random fields can improve the sample variance;
4. **Computational cost.** Our finetuning method turns Aurora into a generative model faster than existing probabilistic finetuning approaches (Weyn et al., 2026). The method only increases the parameter count by 0.64% and scales to the large version of Aurora. Xaurora generates a 15-day forecast in 13 minutes on a single A100 GPU and ensemble members can be parallelised.

2 Background

2.1 Problem statement

Notation. Superscripts denote physical time, subscripts pseudo-time $\tau \in [0, 1]$, lowercase bold ($\boldsymbol{x}$) is data or a sample, uppercase bold ($\boldsymbol{X}_\tau^t$) a stochastic process, and hats ($\hat{\boldsymbol{x}}$) mark generated states.

Let $\boldsymbol{x}^t \in \mathbb{R}^{N_{\text{var}} \times N_{\text{lat}} \times N_{\text{lon}}}$ be the atmospheric state at physical time t, where $N_{\text{var}}, N_{\text{lat}}$, and N_{lon} represent the number of variables, latitude points, and longitude grid points, respectively. Probabilistic weather forecasting aims to sample from $p(\boldsymbol{x}^{t+\Delta t} \mid \boldsymbol{h}^t)$, where $\boldsymbol{h}^t$ is the history of the most recent states, and Δt denotes the temporal interval. This is usually applied autoregressively so that each member (m) conditions on its own generated states, $\hat{\boldsymbol{x}}^{t+k\Delta t,(m)} \sim p(\cdot \mid \hat{\boldsymbol{h}}^{t+(k-1)\Delta t,(m)})$.

In practice, we follow Aurora (Bodnar et al., 2025) and use $N_{\text{var}} = 69$ variables (see Appendix A.1 for a detailed list), as well as a 0.25° resolution grid (*i.e.* $N_{\text{lat}} = 721$ and $N_{\text{lon}} = 1440$). The timestep is set to $\Delta t = 6\,\text{h}$, and we define history as the two most recent states, $\boldsymbol{h}^t := (\boldsymbol{x}^{t-\Delta t}, \boldsymbol{x}^t)$.

Additionally, we strictly focus on modelling atmospheric uncertainty and deliberately ignore other sources of uncertainty, such as that of the data assimilation process generating the initial conditions.

2.2 Stochastic interpolants for autoregressive forecasting

The Stochastic Interpolant (SI) framework learns a continuous-time generative map between two endpoint distributions (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2023). Unlike FM (Lipman et al., 2023) or DDPMs (Ho et al., 2020), SIs do not place a Gaussian assumption on the initial distribution. Setting the endpoints to two consecutive atmospheric states therefore turns the interpolant into a stochastic bridge that enables autoregressive forecasting (Chen et al., 2024; Mücke & Sanderse, 2025; Andrae et al., 2025; Kossaifi et al., 2026) (see Figure 1). We describe the framework directly in this setting. The interpolant connects $\boldsymbol{x}^t$ and $\boldsymbol{x}^{t+\Delta t}$ along a path indexed by a pseudo-time $\tau \in [0, 1]$,

$$\boldsymbol{I}_\tau^t = \alpha_\tau \boldsymbol{x}^t + \beta_\tau \boldsymbol{x}^{t+\Delta t} + \gamma_\tau \boldsymbol{W}_\tau, \tag{1}$$

where $\boldsymbol{W}_\tau \sim \mathcal{N}(\mathbf{0}, \tau\mathbf{I})$ is a standard Wiener process. During training, we sample $\boldsymbol{W}_\tau = \sqrt{\tau}\boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The scalar functions satisfy $\alpha_0 = \beta_1 = 1$ and $\alpha_1 = \beta_0 = \gamma_1 = 0$, so that $\boldsymbol{I}_0^t = \boldsymbol{x}^t$ and $\boldsymbol{I}_1^t = \boldsymbol{x}^{t+\Delta t}$. By Itô's lemma, $\boldsymbol{I}_\tau^t$ solves

$$\mathrm{d}\boldsymbol{I}_\tau^t = \underbrace{\left(\dot{\alpha}_\tau \boldsymbol{x}^t + \dot{\beta}_\tau \boldsymbol{x}^{t+\Delta t} + \dot{\gamma}_\tau \boldsymbol{W}_\tau\right)}_{=:\boldsymbol{R}_\tau^t} \mathrm{d}\tau + \gamma_\tau \mathrm{d}\boldsymbol{W}_\tau. \tag{2}$$

The drift $\boldsymbol{R}_\tau^t$ is not available at inference, as it depends on the target $\boldsymbol{x}^{t+\Delta t}$. It is therefore approximated by a neural network $\boldsymbol{B}_\theta(\cdot, \boldsymbol{h}^t, \tau)$, usually trained for drift matching:

$$\mathcal{L}_{\boldsymbol{B}}(\theta) = \int_0^1 \mathbb{E}_{(\boldsymbol{h}^t, \boldsymbol{x}^{t+\Delta t}) \sim p_{\text{data}}, \boldsymbol{W}_\tau \sim \mathcal{N}(\mathbf{0}, \tau\mathbf{I})} \left[\left\| \boldsymbol{B}_\theta(\boldsymbol{I}_\tau^t, \boldsymbol{h}^t, \tau) - \boldsymbol{R}_\tau^t \right\|_2^2 \right] \mathrm{d}\tau. \tag{3}$$

After training, samples are drawn from $p(\boldsymbol{x}^{t+\Delta t} \mid \boldsymbol{h}^t)$ by integrating the SDE below from $\boldsymbol{X}_0^t = \boldsymbol{x}^t$ with $\boldsymbol{B}_\theta(\cdot, \boldsymbol{h}^t, \tau)$ as drift term,

$$\mathrm{d}\boldsymbol{X}_\tau^t = \boldsymbol{B}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau)\,\mathrm{d}\tau + \gamma_\tau \mathrm{d}\boldsymbol{W}_\tau, \quad \tau \in [0, 1], \quad \boldsymbol{X}_0^t = \boldsymbol{x}^t, \tag{4}$$

and the forecast is the terminal SDE state, $\hat{\boldsymbol{x}}^{t+\Delta t} := \boldsymbol{X}_1^t$. Repeating this autoregressively gives the physical-time rollout (as in Figure 1). The integration scheme and the step count are chosen based on the desired sample fidelity and available compute resources.

Note the distinction between the interpolant ($\boldsymbol{I}_\tau^t$) and the SDE state ($\boldsymbol{X}_\tau^t$). The former is built from the *true* endpoints, which can only be done during training. The latter is generated by the model and fed to the neural network at inference. In practice, in our rollout training algorithm (Section 4.3), we never use the interpolant and train exclusively on generated SDE states.

3 Related work

Deterministic DLWP. The first generation of DLWP models were deterministic (Bi et al., 2023; Pathak et al., 2022; Lam et al., 2023). These models quickly reached global RMSE values comparable to those of NWPs and often outperformed them. However, they suffered from progressively smoother prediction (blurring), as lead times increased, due to their deterministic objective (Subich et al., 2025). Aurora (Bodnar et al., 2025) showed that the vision transformer (ViT) (Dosovitskiy et al., 2021) could be scaled to a billion-parameter atmospheric foundation model. Yu et al. (2026) confirmed that scaling laws (Kaplan et al., 2020) also apply to global DLWP models. Aurora introduced a replay buffer finetuning algorithm, enabling cheap rollout training of very large DLWP models. In contrast to these approaches, we propose a novel generative model, building on Aurora and extending the replay buffer mechanism to SDE-based ensemble models.

Generative DLWP. Many works have proposed generative DLWP models. SEEDS (Li et al., 2023) corrected NWP outputs with a diffusion model at low resolution (2°). GenCast (Price et al., 2023) proposed an autoregressive conditional diffusion model for global weather forecasting at 0.25° resolution. ArchesWeatherGen (Couairon et al., 2024) used diffusion to model the residuals of a separate deterministic model (at 1.5° resolution), which turned out to be challenging since the

residuals have a different distribution on the test set. Mardani et al. (2025) used a similar setup for downscaling (super-resolution). Kossaifi et al. (2026) trained a Diffusion Transformer (DiT) (Peebles & Xie, 2023) at $0.25°$ resolution, under both DDPM and SI frameworks with similar performance. Aich et al. (2026) trained a low-resolution ($1.5°$) diffusion model and explored zero-shot posterior sampling for a variety of weather and climate-relevant tasks. Unlike these methods, we extend and finetune a deterministic foundation model into a generative DLWP model, allowing high-resolution ($0.25°$) for limited training costs.

CRPS training. Another approach to build a probabilistic DLWP consists of optimising ensemble forecasts with a probabilistic loss, *e.g.* CRPS Gneiting & Raftery (2007). Lang et al. (2024b) showed that an ensemble size of 2 was sufficient, roughly doubling the compute and memory requirements over deterministic training. CRPS training requires stochastic networks, which is often achieved by injecting noise in FiLM (Perez et al., 2018) layers (Lang et al., 2024b; Alet et al., 2025; Kossaifi et al., 2026). Cachay et al. (2026) showed that dropout (Hinton et al., 2012) could also be used. While inference is cheaper with CRPS-trained models, they still suffer from overly smooth predictions (Lang et al., 2024b). They are also limited to their training task and cannot be cheaply expanded in the same way generative models can (*e.g.* posterior sampling (Chen et al., 2025; Aich et al., 2026) or physics-constrained generation (Jacobsen et al., 2024; Cheng et al., 2025; Utkarsh et al., 2025)).

Probabilistic extensions of Aurora. Two probabilistic extensions of Aurora have recently been proposed (Ozdemir et al., 2026; Weyn et al., 2026). Both models use CRPS training and focus on extending the foundational aspect of Aurora by increasing the number of variables and modalities. Our work provides a complementary perspective exploring generative finetuning.

4 Denoising Stochastic Interpolants and Xaurora

In this section we introduce a novel generative model, Denoising Stochastic Interpolants (DSI), that is robust in high-dimensional pixel spaces and well-suited for finetuning deterministic priors (Section 4.1). Furthermore, we design a probabilistic SDE-rollout training to capture the uncertainty emerging from the forecasts (Section 4.3). Finally, we build *Xaurora* by applying this methodology to an atmospheric foundation model (Section 4.4).

4.1 Denoising Stochastic Interpolants

Diffusion models can be expressed as three equivalent formulations, $\boldsymbol{\epsilon}$-, $\boldsymbol{x}$-, and $\boldsymbol{v}$-*prediction*, respectively predicting the noise added to the clean signal, the clean signal itself, and the velocity field from the noised signal to the clean signal (Ho et al., 2020; Salimans & Ho, 2022; Li & He, 2026). We show that the same reparameterisation trick can be applied to SIs. Namely, we define $\boldsymbol{W}$-, $\boldsymbol{x}_1$- and $\boldsymbol{B}$-prediction as analogues of the diffusion parameterisations (more details in Appendix A.5). $\boldsymbol{x}_1$-prediction is the SI equivalent of $\boldsymbol{x}$-prediction, which in our temporal setting corresponds to predicting $\boldsymbol{x}^{t+\Delta t}$. In diffusion, this parameterisation is often referred to as *denoising*, hence we call this method *Denoising Stochastic Interpolants* (DSI).

Let $\boldsymbol{D}_\theta$ be a *denoiser* model predicting the terminal state $\boldsymbol{x}^{t+\Delta t}$ from an arbitrary SDE state $\boldsymbol{X}_\tau^t$,

$$\boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau) \approx \mathbb{E}[\boldsymbol{x}^{t+\Delta t} \mid \boldsymbol{X}_\tau^t, \boldsymbol{h}^t]. \tag{5}$$

The drift in Equation (4) can be rewritten as (see Appendix A.5 for derivation details)

$$\boldsymbol{B}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau) = \dot{\alpha}_\tau \boldsymbol{x}^t + \dot{\beta}_\tau \boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau) + \frac{\dot{\gamma}_\tau}{\gamma_\tau}\big[\boldsymbol{X}_\tau^t - \alpha_\tau \boldsymbol{x}^t - \beta_\tau \boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau)\big], \tag{6}$$

with $\boldsymbol{x}^t$ becoming a generated state when performing autoregressive rollouts ($\hat{\boldsymbol{x}}^{t+k\Delta t}$, $k > 0$). A common choice of γ_τ is $1 - \tau$. To avoid numerical instabilities at $\tau = 1$, we clip the denominator of the last term to 0.05. During training, we introduce the denoising loss instead of Equation (3),

$$\mathcal{L}_{\boldsymbol{D}}(\theta) = \int_0^1 \mathbb{E}_{(\boldsymbol{h}^t, \boldsymbol{x}^{t+\Delta t}) \sim p_{\text{data}}, \boldsymbol{W}_\tau \sim \mathcal{N}(\boldsymbol{0}, \tau \boldsymbol{I})} \big[\big\| \boldsymbol{D}_\theta(\boldsymbol{I}_\tau^t, \boldsymbol{h}^t, \tau) - \boldsymbol{x}^{t+\Delta t} \big\|_2^2 \big] \mathrm{d}\tau. \tag{7}$$

Inspired by Li & He (2026), we also show that the denoiser loss can be equivalently parameterised in $\boldsymbol{B}$ and $\boldsymbol{W}$-space simply by introducing a τ-dependent weight (more details in Appendix A.5).

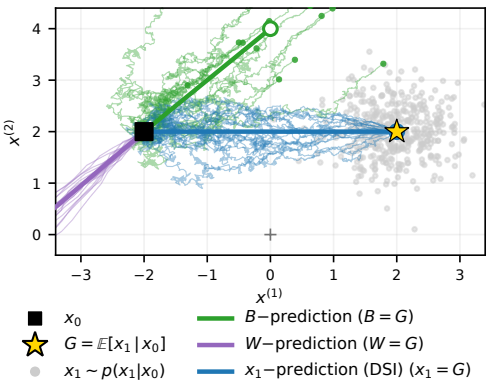

Figure 2: **2D toy example: only DSI preserves an ideal prior.** In each parameterisation (Table 5), we replace the network with an ideal prior $\boldsymbol{G}(\boldsymbol{x}_0) = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0]$ and integrate Equation (4) from a common initial state (black square), no training. Only DSI recovers the true mean (gold star) as its network models $\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0]$, exactly what $\boldsymbol{G}$ computes. The deterministic prior has no spread, hence the SDE trajectories (thin lines, thick line is the average) land on the expectation. Hence, finetuning (Section 4.2) aims to capture the spread and maintain the skill. We design our algorithm (Section 4.3) for that purpose and apply it to an atmospheric foundation model (Section 4.4).

4.2 Denoising Stochastic Interpolants from a Pretrained Prior

Let $\boldsymbol{G}_\phi$ be a pretrained deterministic prior with whose parameters ϕ are a subset of θ, such that

$$\boldsymbol{G}_\phi(\boldsymbol{h}^t) \approx \mathbb{E}[\boldsymbol{x}^{t+\Delta t} \mid \boldsymbol{h}^t]. \tag{8}$$

Our objective is to finetune a stochastic interpolant model from this prior, providing a warm-start to the optimisation. In this context, DSI is the best of the three parameterisations discussed above. We assess this intuition with a toy example shown in Figure 2. The mathematical motivation for the general case in arbitrary dimensions is presented in Appendix A.5.2.

Warm start. In Figure 2, only DSI reaches the correct endpoint. That is, without any finetuning, initialising the denoiser such that $\boldsymbol{D}_{\theta^{\text{init}}} \equiv \boldsymbol{G}_\phi$ (where $\equiv$ denotes functional equivalence) will fully maintain the capabilities of the pretrained prior, while these capabilities are (partially) lost with the $\boldsymbol{B}$- and $\boldsymbol{W}$-prediction parameterisations. These results hold for $\dot{\beta}_0 \neq 0$ as discussed in Appendix A.6.4. Hence, we use $\beta_\tau = \tau$.

Consistent network target. In DSI, the denoiser and the pretrained prior are trained against the same kind of target, a conditional expectation of the terminal state $\boldsymbol{x}^{t+\Delta t}$. This is not the case for the other parameterisations whose targets typically involve a combination of states and random fields, which introduces new modalities to the underlying model, potentially degrading convergence.

Curse of dimensionality. Li & He (2026) showed that $\boldsymbol{x}$-prediction was preferable for diffusion in high-dimensional spaces. That is because predicting noise ($\boldsymbol{\epsilon}$) or noisy quantities ($\boldsymbol{v}$) is subject to the *curse of dimensionality*. In contrast, $\boldsymbol{x}$-prediction remains on the data manifold, achieving higher performance in those cases. Given the very high dimensionality of atmospheric states, we hypothesise that these conclusions hold.

As long as $\boldsymbol{D}_{\theta^{\text{init}}} \equiv \boldsymbol{G}_\phi$ (functionally equivalent), these observations carry over to the realistic scenario from Equation (8). In that case, DSI finetuning serves two purposes; (1) modelling the spread around the prior and (2) surpassing the skill of the prior through ensemble predictions.

For completeness, we note that in the case of residual prediction, *i.e.* $\boldsymbol{G}_\phi(\boldsymbol{h}^t) \approx \mathbb{E}[\boldsymbol{x}^{t+\Delta t} - \boldsymbol{x}^t \mid \boldsymbol{h}^t]$, another common formulation in autoregressive state prediction, $\boldsymbol{B}$-prediction, is likely preferable (see Appendix A.5.2). Nevertheless, the network still targets a noisy quantity and could be less robust to the curse of dimensionality, following the arguments of Li & He (2026).

4.3 SDE-Rollout Replay Buffer Finetuning with Probabilistic Loss

Rollout training, or training a model on its own autoregressive forecasts, is crucial for long term performance in DLWP (Lam et al., 2023; Lang et al., 2024a;b). However, traditional approaches (Lam et al., 2023) are not tractable with generative models, as each physical step involves solving an entire SDE (nested rollout, see Figure 1). We solve this by blending the physical and SDE rollouts together. We take inspiration from Bodnar et al. (2025), who emulate rollout training using a replay buffer. We extend this procedure to nested rollouts, enabling rollout training with negligible computational overhead compared to single-step optimisation.

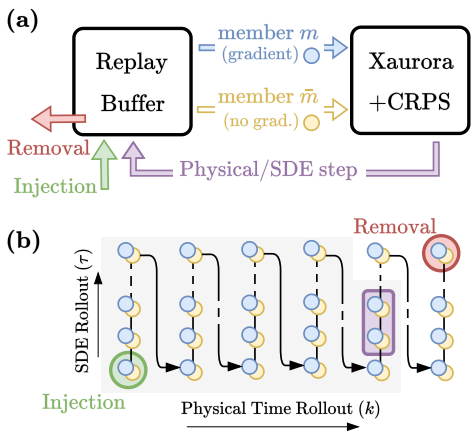

Figure 3: **SDE-rollout training with single-backpropagation CRPS loss. (a)** Two members are sampled from the buffer and fed to the network, only member m is backpropagated. Both advance one step (physical step if τ reaches 1, SDE step otherwise), and are pushed back. Fresh samples are randomly injected, old (FIFO) or finished rollouts are removed. **(b)** The buffer tracks pairs of members (yellow and blue circles) throughout nested rollouts (τ, k). Both members start from the same initial condition, but receive different noise samples. A training iteration costs the same at every depth (2 forward, 1 backward, purple box), unlike regular rollout training that require full nested rollouts (shown in grey).

In the original SI (Albergo & Vanden-Eijnden, 2023), the drift approximates $\mathbb{E}[\boldsymbol{R}_\tau^t \mid \boldsymbol{X}_\tau^t, \boldsymbol{h}^t]$ and solving the SDE (Equation (4)) recovers samples from $p(\boldsymbol{x}^{t+\Delta t} \mid \boldsymbol{h}^t)$. We instead propose to train a sampler end-to-end by directly optimising SDE rollouts with a proper score, the fair CRPS loss (Lang et al., 2024b). This results in a different minimiser, corresponding to a probability flow with increased variance. Since our ensemble is generally underdispersive (Section 5.1), this is generally beneficial to performance (see ablation experiments in Section 5.3). Given the ground truth $\boldsymbol{x}^{t+\Delta t}$, a pair of members $(\hat{\boldsymbol{x}}^{(m)}, \hat{\boldsymbol{x}}^{(\bar{m})})$, and $\|\cdot\|_\omega$ as the variable-weighted ℓ_1-norm (with weights ω), our fair CRPS loss is given by:

$$\ell_{\text{fCRPS}}(\boldsymbol{x}^{t+\Delta t}, \hat{\boldsymbol{x}}^{(m)}, \hat{\boldsymbol{x}}^{(\bar{m})}; \theta) := \underbrace{\frac{1}{2}\left(\sum_{i \in \{m, \bar{m}\}} \left\| \hat{\boldsymbol{x}}^{(i)} - \boldsymbol{x}^{t+\Delta t} \right\|_\omega\right)}_{\text{skill term } (\downarrow)} - \underbrace{\frac{1}{2}\left\| \hat{\boldsymbol{x}}^{(m)} - \hat{\boldsymbol{x}}^{(\bar{m})} \right\|_\omega}_{\text{spread term } (\uparrow)}. \quad (9)$$

The first term is the *skill term*, measuring the prediction error, and is equivalent to the sum of the MAE of each member. The second term is the *spread term*, measuring the variance across samples. Unlike other methods (Lang et al., 2024b; Alet et al., 2025; Kossaifi et al., 2026; Cachay et al., 2026), we propose to supervise only one randomly selected member when computing the loss, resulting in the same local minima while minimising the computational cost. The process is visualised in Figure 3, and more details are given in Appendix A.6. We finetuned Xaurora for only 10% of the steps used to pretrain Aurora, which takes 42 hours on 16 A100 GPUs, roughly 20× less GPU hours than the probabilistic Aurora model by Weyn et al. (2026).

4.4 Denoiser model

The denoiser of Xaurora is based on Aurora (Bodnar et al., 2025), a SOTA deterministic atmospheric foundation model. By initialising $\boldsymbol{G}_\phi$ to Aurora-S (the small version of Aurora), the approach discussed in Section 4.2 can be used for weather forecasting. As Aurora predicts $\boldsymbol{x}^{t+\Delta t}$ from $\boldsymbol{h}^t$, we make minimal architectural changes (see Figure 6 in Appendix A.3) to accommodate the pseudo-time τ and the SDE state $\boldsymbol{X}_\tau^t$ according to Equation (5). These changes increase the parameter count by 0.64% and ensure $\boldsymbol{D}_{\theta^{\text{init}}} \equiv \boldsymbol{G}_\phi$ with *zero-initialisation* (Zhang et al., 2023).

Conditioning on $\boldsymbol{X}_\tau^t$. We create new surface and atmospheric patch embeddings layers in the encoder dedicated to the SDE state. We sum these embeddings with those of $\boldsymbol{h}^t$.

Conditioning on τ. The pseudo-time is embedded with a Fourier Expansion (Equation (10)) and a zero-initialised MLP. The resulting vector is added to the pre-existing lead time embedding and fed to AdaLN layers (Perez et al., 2018; Peebles & Xie, 2023) in the Swin Transformer (Liu et al., 2021).

LoRA. The Swin Transformer is entirely frozen and finetuned with Low Rank Adaptation (LoRA) (Hu et al., 2021) with rank 8.

Network preconditioning. We adapt the preconditioning from Karras et al. (2022) to DSI. This ensures that the model inputs, particularly $\boldsymbol{X}_\tau^t$, are approximately zero-mean unit-variance on average. This is entirely parameter-free (more details in Appendix A.3.2).

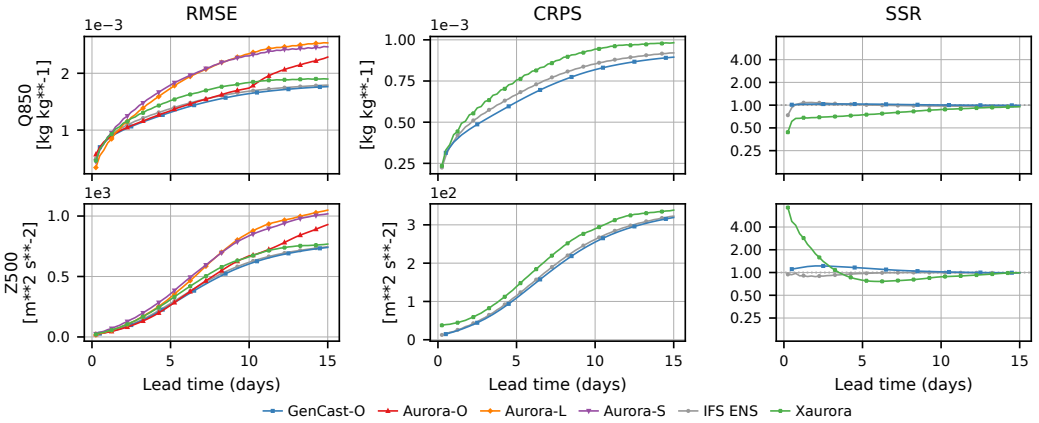

Figure 4: **Global medium-range forecast skill in 2022.** From left to right, columns represent RMSE (uRMSE for ensemble models), fair CRPS (ensemble only), and SSR (ensemble only). Rows represent different variables; Q850 is a proxy for mesoscale variability, Z500 for large-scale circulation and synoptic-scale variability. Xaurora's ensemble mean unbiased RMSE outperforms Aurora-S/L on most lead times and is competitive with operational models. All models are well calibrated at 15 days (SSR$\approx$1). Xaurora's ensemble is generally underdispersive (SSR$<$1). Z500 is overdispersive (SSR$>$1) until approximately 3 days.

5 Experiments

5.1 Medium-range weather forecasting

We use WeatherBench (Rasp et al., 2024) to benchmark Xaurora against SOTA ensemble systems, using the year 2022 as the test set. Metrics are defined in Appendix A.4.1. We compare Xaurora against the three Aurora versions: Aurora-Small (Aurora-S), Aurora-Large (Aurora-L), and Operational Aurora (Aurora-O). Furthermore, we compare against Operational GenCast[1] (GenCast-O (Price et al., 2023)), which serves to represent DLWP, and the SOTA NWP system, IFS ENS.

Qualitative results. Xaurora produces high-quality forecasts that are realistic compared to ERA5 samples (see Figures 40, 41 and 42). The model generates multi-scale features, from planetary waves in geopotential fields (Z) to grid-scale structures in the moisture fields (Q). After 3 days, individual Xaurora members remain close to Aurora-S, whereas Xaurora diverges from Aurora-S after 10 days, indicating that Xaurora benefits from the prior without being overly constrained.

Ensemble mean performance. The RMSE panels in Figure 4 show that the ensemble mean of Xaurora is close to that of the operational ensemble forecasting systems (IFS ENS, Aurora-O, GenCast-O). The performance gap is minimal at initialisation and the error curves follow similar trends. Xaurora also outperforms its prior, Aurora-S, beyond 1 day and Aurora-L beyond 3 days (see also Figures 38 and 39). Beyond 7 to 12 days, Xaurora achieves a lower error than Aurora-O, which has only been finetuned for 10-day rollouts. The error reduction over Aurora models cannot be entirely attributed to our method since averaging ensembles provides a statistical advantage (which is also partly the motivation for using ensembles over deterministic models). An ensemble size of 2 is already sufficient to outperform the prior (see B.3.3). More RMSE curves can be found in Figure 29.

Probabilistic performance. The CRPS scores of Xaurora closely follow those of the operational models (see Figures 4 and 30). Note that we use the fair CRPS since the ensemble sizes differ between systems. At 6 h, the CRPS of Z500 is higher than other models. Overall, our ensemble is slightly underdispersive (SSR$<$1) and well calibrated at 15 days (see Figures 4 and 31). However, on short lead times (6 h to 3 days), synoptic-scale variables (Z, T, 2T, MSL) are overdispersive, whereas mesoscale variables (Q, U, V) are underdispersive[2]. Finally, Figure 28 shows that Xaurora consistently outperforms a naive probabilistic version of Aurora.

[1] Among the ensemble DLWP models available on WeatherBench, we choose GenCast because it is widely considered the reference in probabilistic and generative DLWP.

[2] Synoptic-scale variables are mostly influenced by synoptic-scale variability, *i.e.* large scale phenomena such as the jet stream, whereas mesoscale ones are driven by small scale processes such as storms and convection.

5.2 SPECTRAL CALIBRATION ANALYSIS

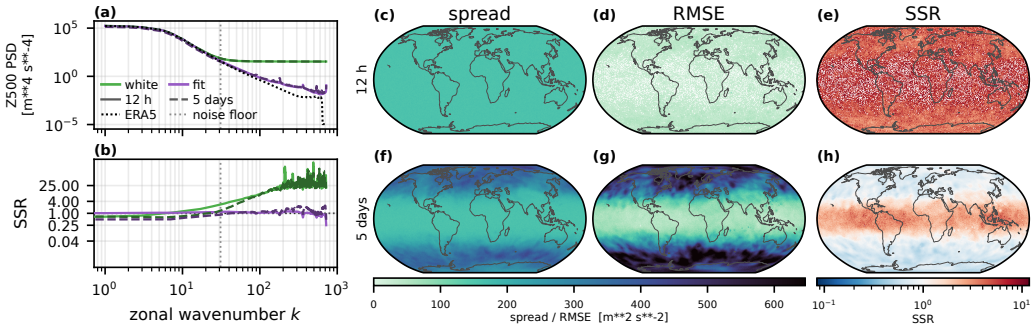

Figure 5: **Spectral calibration and spatial structure.** **(a)** Average Z500 zonal Power Spectral Density (PSD) at 12 hours (solid) and 5 days (dashed) with white noise (green) and error-fitted noise (purple, see Appendix B.1.5), compared to ERA5 (dotted). The vertical dotted line indicates the cut-off wavenumber ($k = 31$; defined as the first k where the power is twice as large as that of ERA5). **(b)** Cumulative spectral SSR at 12 hours and 5 days, can be interpreted as *SSR up to wavenumber k*. **(c-e)** Spatial spread, RMSE, and SSR for Z500 at 12 hours (with white noise). **(f-h)** Same at 5 days. The noise floor is visible in the spectral plots (k>31) and the lack of tropical structure in (c) and (f).

We observe that all variables show a flat (or flattening) Power Spectral Density (PSD) after a specific cut-off wavenumber; the PSD panel of Figure 5 places it at $k = 31$ for Z500. The cut-off wavenumbers and the corresponding power are lead-time independent. We call this phenomenon the *noise floor*, as no structure can be fully recovered beyond this threshold. Mesoscale variables typically have larger cut-off wavenumbers than synoptic-scale variables (see Figures 32 and 33 for more variables and comparison with Aurora-S). From a weather forecasting point of view, those high wavenumbers are far less relevant since they have limited predictability on medium range timescales.

The ensemble is underdispersive on large scales (low wavenumbers) regardless of the variable (see the SSR panels in Figures 5 and 12). At 12 h lead time, Z500 and other synoptic-scale variables are overdispersive on small scales that dominate the global SSR. This explains their global overdispersion on short lead times as a pure product of high wavenumber activity. This can be related to the noise floor; on short lead times, where the ensemble mean RMSE is low, the high frequency power required to calibrate the ensemble would be below the noise floor. Hence, small scales are overdispersed, and the global SSR is greater than 1. Spatially, it induces overdispersion in regions of low activity, *e.g.* in the tropics for Z500 (see the 5-day SSR map in Figure 5). Spatial maps of other variables, rank histograms, and pooled SSR curves suggest similar results (Figures 34, 35, 36 and 37).

Our investigations revealed that the noise floor is invariant to the duration of training, the generative model, the SDE solver, the Aurora version used as prior, the optimisation method, the CRPS loss, and whether the denoiser is finetuned or trained from scratch (see Appendix B.2). Although the network architecture could also play a role, we hypothesise that the diffusion term in the SDE has a major influence on the noise floor. To examine this hypothesis, we conduct a small-scale experiment in which a noise model is fitted to the PSD of the error. The experiment confirms our hypothesis, as the fitted noise correctly calibrates the SSR (Figure 5), but costs long-term RMSE (see Appendix B.1). The increase in RSME likely comes from the new diffusion term perturbing the predictable large scales at inference. Using similar approaches during training could instead prevent the noise floor.

5.3 ABLATIONS

We conduct a suite of ablation studies to measure the benefits of our optimisation method, the DSI parameterisation, and whether Xaurora scales to using Aurora-L as the prior. The results confirm the effectiveness of our method for probabilistic weather forecasting and are summarised in Table 1.

Optimisation. Training with MAE results in better uRMSE (up to 3.1%), but worse in CRPS (up to 15.5%). The curriculum and replay buffer both improve uRMSE further. Single-step + MAE performs close to Xaurora; only CRPS Q850 is significant. A 15-day evaluation shows that Xaurora is significantly better across variables and metrics on longer leads (see Appendix B.3.2).

Table 1: **Ablation experiments results.** Skill scores against Xaurora, $100 \cdot (1 - \mathcal{M}_{\text{model}}/\mathcal{M}_{\text{Xaurora}})$: positive values are better than the reference. The best values are underlined. Methods are detailed in Appendix B.3.1. † marks differences that are not significant at the 95% level (see Appendix A.4.2).

		5-day uRMSE		5-day CRPS	
		Z500	Q850	Z500	Q850
Xaurora (Replay buf. + CRPS + Aurora-S)		309.9	1.52×10^{-3}	<u>139.4</u>	<u>7.55×10^{-4}</u>
Aurora-S		-15.4%	-18.2%	N/A	N/A
Optim.	Single-step + MAE	+0.0%†	+0.1%†	-1.3%†	-6.5%
	Curriculum + MAE	<u>+3.1%</u>	+1.0%	-4.1%	-15.5%
	Replay buf. + MAE	+1.4%†	<u>+2.5%</u>	-4.7%	-14.9%
Param.	$\boldsymbol{x}_1$-pred. + DDPM	-85.5%	-31.8%	-116.5%	-45.3%
	$\boldsymbol{B}$-pred.	-89.7%	-33.3%	-1315.1%	-205.2%
	$\boldsymbol{W}$-pred. (Training diverges)	NaN	NaN	NaN	NaN
Xaurora (Single-step + MAE + Aurora-L)		<u>325.9</u>	<u>1.65×10^{-3}</u>	153.5	8.71×10^{-4}
Aurora-L		-0.6%†	-5.1%	N/A	N/A

Parameterisation. $\boldsymbol{B}$ and $\boldsymbol{W}$-pred cannot be warm started correctly (see the toy example in Figure 2) and perform poorly ($\boldsymbol{W}$-pred diverges, hence the *NaN* values). The real competitor is $\boldsymbol{x}_1$-pred + DDPM, which also massively underperforms. Appendix B.2.1 shows that this is not due to insufficient integration steps. We note that the replay buffer training is softly restricted to 10 steps, potentially too few for DDPM, but not for DSI, which again highlights its efficiency.

Prior. Initialising the denoiser to Aurora-L reproduces the core result, *i.e.* the ensemble outperforms the prior, even when training with single step + MAE and without having fully converged. Based on the optimisation results, we expect significantly better results should the training be scaled and made probabilistic. This was not pursued further due to the high computational cost.

6 Conclusion

In this work, we introduced Denoising Stochastic Interpolants (DSI), a new generative model, and a probabilistic rollout training algorithm for SDE-based autoregressive state prediction. This setup enables warm starts from pretrained deterministic priors in high-dimensional settings. These novelties allowed us to build Xaurora by efficiently fine-tuning Aurora, a deterministic atmospheric foundation model. Our 15-day ensemble forecasts are generated in 13 minutes and approach the performance of SOTA probabilistic weather forecasting systems.

Limitations and future work. Though high frequencies are less important than large scales for operational weather forecasting, the persistent noise floor highlighted by our analysis is the main limitation of Xaurora. We identified the major influence of the noise model on the spectral properties of the forecasts, which inference-only interventions can only partially calibrate. Further research can focus on building random fields that are better suited to the spectral properties of the task. Stochastic Interpolants technically do not require the noise to follow a Gaussian distribution. Hence, combining SI with different noise models should be possible, although the geometrical aspect of the problem remains fairly unexplored. A more ambitious direction would consist in learning the diffusion term together with the denoiser. This would involve optimising the SDE together with the model parameters; a non-trivial task to which our probabilistic SDE-rollout training algorithm could be particularly well-suited. Other relevant directions include distillation for sampling efficiency (Song et al., 2023; Boffi et al., 2024; Stock et al., 2026) and physics-constrained modelling (Jacobsen et al., 2024; Cheng et al., 2025; Utkarsh et al., 2025).

Broader impact. Finetuning Xaurora and generating forecasts has a relatively low computational cost, making skilful probabilistic weather forecasting more accessible to institutes with limited compute (*e.g.* academia or national meteorological offices). Our method is also scalable to larger models or output spaces and is expected to transfer to other problems beyond weather forecasting.

AI USE STATEMENT

This work used generative AI to help develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, and interpret results. Furthermore, generative AI was used to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, as well as general brainstorming. The LLM-generated code, interpretations, and suggestions were always subjected to human verification.

AI was not used to generate synthetic data sets, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation, clean and reformat dataset, or support qualitative and thematic data analysis.

ETHICS STATEMENT

This work uses only publicly available reanalysis and operational forecast data (ERA5, HRES-T0 and IFS ENS) and involves no human subjects or personally identifiable information. By turning an existing deterministic foundation model into a probabilistic ensemble model at a small fraction of the cost of training a generative forecaster from scratch, our method lowers the barrier to ensemble forecasting for institutions without large compute budgets, which we consider its principal societal benefit. Xaurora is nonetheless a research prototype: it inherits the biases and uneven observational coverage of its ERA5 training data, so its skill is likely lower over data-sparse regions, and it remains underdispersive at short lead times for some variables, and it should therefore not be used for operational warnings or other high-stakes decisions without independent validation by forecasters.

REPRODUCIBILITY STATEMENT

This work is reproducible. We use an appropriate software stack for version control, experiment logging, and configuration tracking. The source code is provided in the supplementary materials and the repository will be made available upon acceptance. All data used in this study were downloaded from WeatherBench. This represents tens of terabytes of storage.

REFERENCES

Michael Aich, Andreas Fürst, Florian Sestak, Carlos Ruiz-Gonzalez, Niklas Boers, and Johannes Brandstetter. WIND: Weather Inverse Diffusion for Zero-Shot Atmospheric Modeling, May 2026. URL http://arxiv.org/abs/2602.03924. arXiv:2602.03924 [cs.LG].

Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants, March 2023. URL http://arxiv.org/abs/2209.15571. arXiv:2209.15571 [cs].

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions, November 2023. URL http://arxiv.org/abs/2303.08797. arXiv:2303.08797 [cs].

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, June 2025. URL http://arxiv.org/abs/2506.10772. arXiv:2506.10772 [cs.LG] version: 1.

Martin Andrae, Erik Wikingsson, So Takao, Tomas Landelius, and Fredrik Lindsten. DAISI: Data Assimilation with Inverse Sampling using Stochastic Interpolants, 2025. URL https://arxiv.org/abs/2512.00252. Version Number: 4.

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, July 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06185-3. URL https://www.nature.com/articles/s41586-023-06185-3.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, Jayesh K. Gupta, Kit Thambiratnam, Alexander T. Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. A foundation model for the Earth system. *Nature*, 641 (8065):1180–1187, May 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09005-y. URL https://www.nature.com/articles/s41586-025-09005-y.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow Map Matching, June 2024. URL http://arxiv.org/abs/2406.07507. arXiv:2406.07507 [cs] version: 1.

Salva Rühling Cachay, Duncan Watson-Parris, and Rose Yu. U-Cast: A Surprisingly Simple and Efficient Frontier Probabilistic AI Weather Forecaster, May 2026. URL http://arxiv.org/abs/2604.09041. arXiv:2604.09041 [cs.LG].

Siyi Chen, Yixuan Jia, Qing Qu, He Sun, and Jeffrey A. Fessler. FlowDAS: A Stochastic Interpolant-based Framework for Data Assimilation, June 2025. URL http://arxiv.org/abs/2501.16642. arXiv:2501.16642 [eess.SP].

Yifan Chen, Mark Goldstein, Mengjian Hua, Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Probabilistic Forecasting with Stochastic Interpolants and Föllmer Processes, August 2024. URL http://arxiv.org/abs/2403.13724. arXiv:2403.13724 [cs].

Chaoran Cheng, Boran Han, Danielle C. Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W. Mahoney, and Yuyang Wang. Gradient-Free Generation for Hard-Constrained Systems, March 2025. URL http://arxiv.org/abs/2412.01786. arXiv:2412.01786 [cs.LG].

Guillaume Couairon, Renu Singh, Anastase Charantonis, Christian Lessig, and Claire Monteleoni. ArchesWeather & ArchesWeatherGen: a deterministic and generative model for efficient ML weather forecasting, December 2024. URL http://arxiv.org/abs/2412.12971. arXiv:2412.12971 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. URL http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs].

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, October 2016. URL http://arxiv.org/abs/1506.02142. arXiv:1506.02142 [stat.ML].

Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 0162-1459. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437. _eprint: https://doi.org/10.1198/016214506000001437.

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 De 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 de 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. ISSN 1477-870X. doi: 10.1002/qj.3803.

URL https://onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qj.3803.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors, July 2012. URL http://arxiv.org/abs/1207.0580. arXiv:1207.0580 [cs.NE].

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020. URL http://arxiv.org/abs/2006.11239. arXiv:2006.11239 [cs.LG].

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2790–2799. PMLR, May 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs.CL].

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep Networks with Stochastic Depth, July 2016. URL http://arxiv.org/abs/1603.09382. arXiv:1603.09382 [cs.LG].

Christian Jacobsen, Yilin Zhuang, and Karthik Duraisamy. CoCoGen: Physically-Consistent and Conditioned Score-based Generative Models for Forward and Inverse Problems, October 2024. URL http://arxiv.org/abs/2312.10527. arXiv:2312.10527 [cs.LG].

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General Perception with Iterative Attention, March 2021. URL https://arxiv.org/abs/2103.03206v2.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, January 2020. URL http://arxiv.org/abs/2001.08361. arXiv:2001.08361 [cs.LG].

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the Design Space of Diffusion-Based Generative Models, October 2022. URL http://arxiv.org/abs/2206.00364. arXiv:2206.00364 [cs.CV].

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, March 2024. URL http://arxiv.org/abs/2311.07222. arXiv:2311.07222.

Jean Kossaifi, Nikola Kovachki, Morteza Mardani, Daniel Leibovici, Suman Ravuri, Ira Shokar, Edoardo Calvello, Mohammad Shoaib Abbas, Peter Harrington, Ashay Subramaniam, Noah Brenowitz, Boris Bonev, Wonmin Byeon, Karsten Kreis, Dale Durran, Arash Vahdat, Mike Pritchard, and Jan Kautz. Demystifying Data-Driven Probabilistic Medium-Range Weather Forecasting, January 2026. URL http://arxiv.org/abs/2601.18111. arXiv:2601.18111 [cs.LG] version: 1.

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. GraphCast: Learning skillful medium-range global weather forecasting, August 2023. URL http://arxiv.org/abs/2212.12794. arXiv:2212.12794 [physics].

Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana C. A. Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, Zied Ben Bouallègue, Ana Prieto Nemesio, Peter D. Dueben, Andrew Brown, Florian Pappenberger, and

Florence Rabier. AIFS - ECMWF's data-driven forecasting system, June 2024a. URL http://arxiv.org/abs/2406.01465. arXiv:2406.01465 [physics].

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, December 2024b. URL http://arxiv.org/abs/2412.15832. arXiv:2412.15832 [physics].

Lizao Li, Rob Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. SEEDS: Emulation of Weather Forecast Ensembles with Diffusion Models, October 2023. URL http://arxiv.org/abs/2306.14066. arXiv:2306.14066 [cs.LG].

Tianhong Li and Kaiming He. Back to Basics: Let Denoising Generative Models Denoise, January 2026. URL http://arxiv.org/abs/2511.13720. arXiv:2511.13720 [cs].

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2011.00777.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2011.00777.x.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023. URL http://arxiv.org/abs/2210.02747. arXiv:2210.02747 [cs.LG].

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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, Montreal, QC, Canada, October 2021. IEEE. ISBN 978-1-6654-2812-5. doi: 10.1109/ICCV48922.2021.00986. URL https://ieeexplore.ieee.org/document/9710580/.

E. N. Lorenz. Atmospheric predictability experiments with a large numerical model. *Tellus*, 34 (6), January 1982. ISSN 3035-9554. doi: 10.3402/tellusa.v34i6.10836. URL https://tellusjournal.org/articles/10.3402/tellusa.v34i6.10836.

Edward N. Lorenz. The predictability of a flow which possesses many scales of motion. *Tellus*, 21(3): 289–307, 1969. ISSN 2153-3490. doi: 10.1111/j.2153-3490.1969.tb00444.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2153-3490.1969.tb00444.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2153-3490.1969.tb00444.x.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs.LG].

Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Mohammad Amin Nabian, Tao Ge, Akshay Subramaniam, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Residual corrective diffusion modeling for km-scale atmospheric downscaling. *Communications Earth & Environment*, 6(1):124, February 2025. ISSN 2662-4435. doi: 10.1038/s43247-025-02042-5. URL https://www.nature.com/articles/s43247-025-02042-5.

Nikolaj T. Mücke and Benjamin Sanderse. Physics-aware generative models for turbulent fluid flows through energy-consistent stochastic interpolants, April 2025. URL http://arxiv.org/abs/2504.05852. arXiv:2504.05852 [cs].

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. ClimaX: A foundation model for weather and climate, December 2023. URL http://arxiv.org/abs/2301.10343. arXiv:2301.10343 [cs].

Firat Ozdemir, Yun Cheng, Salman Mohebi, Fanny Lehmann, Simon Adamov, Zhenyi Zhang, Leonardo Trentini, Dana Grund, Oliver Fuhrer, Torsten Hoefler, Siddhartha Mishra, Sebastian Schemm, Benedikt Soja, and Mathieu Salzmann. Earth System Foundation Model (ESFM): A

unified framework for heterogeneous data integration and forecasting, April 2026. URL http://arxiv.org/abs/2605.00850. arXiv:2605.00850 [physics].

T. N. Palmer, Roberto Buizza, F. Doblas-Reyes, T. Jung, Martin Leutbecher, G. J. Shutts, M. Steinheimer, and Antje Weisheimer. Stochastic parametrization and model uncertainty, October 2009. URL https://www.ecmwf.int/node/11577. Issue: 598 Pages: 42 Publication Title: ECMWF Technical Memoranda.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators, February 2022. URL http://arxiv.org/abs/2202.11214. arXiv:2202.11214 [physics].

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers, March 2023. URL http://arxiv.org/abs/2212.09748. arXiv:2212.09748 [cs].

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v32i1.11671. URL https://ojs.aaai.org/index.php/AAAI/article/view/11671.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Timo Ewalds, Andrew El-Kadi, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. GenCast: Diffusion-based ensemble forecasting for medium-range weather, December 2023. URL http://arxiv.org/abs/2312.15796. arXiv:2312.15796 [physics].

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):140:5485–140:5551, January 2020. ISSN 1532-4435. URL https://dl.acm.org/doi/10.5555/3455716.3455856.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russel, 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, January 2024. URL http://arxiv.org/abs/2308.15560. arXiv:2308.15560 [physics].

Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models, June 2022. URL http://arxiv.org/abs/2202.00512. arXiv:2202.00512 [cs.LG].

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252. PMLR, July 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Jason Stock, Troy Arcomano, and Rao Kotamarthi. Swift: an autoregressive consistency model for efficient weather forecasting. *Machine Learning: Earth*, 2(2):025004, July 2026. ISSN 3049-4753. doi: 10.1088/3049-4753/ae802a. URL https://doi.org/10.1088/3049-4753/ae802a.

Christopher Subich, Syed Zahid Husain, Leo Separovic, and Jing Yang. Fixing the Double Penalty in Data-Driven Weather Forecasting Through a Modified Spherical Harmonic Loss Function, January 2025. URL http://arxiv.org/abs/2501.19374. arXiv:2501.19374 [cs] version: 1.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-Constrained Flow Matching: Sampling Generative Models with Hard Constraints, November 2025. URL http://arxiv.org/abs/2506.04171. arXiv:2506.04171 [cs.LG].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs] version: 5.

Jonathan Weyn, Zekun Ni, Amit Misra, Will Fein, Haiyu Dong, Wessel P. Bruinsma, Richard E. Turner, Matt Corey, Kit Thambiratnam, Kevin White, and Kenji Takeda. Aurora 1.5: Fine-Tuning a Foundation Model for Medium-Range Ensemble Weather Prediction. July 2026. URL https://www.microsoft.com/en-us/research/publication/aurora-1-5-fine-tuning-a-foundation-model-for-medium-range-ensemble-weather-prediction/.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://papers.nips.cc/paper_files/paper/2014/hash/532a2f85b6977104bc93f8580abbb330-Abstract.html.

Yuejiang Yu, Langwen Huang, Alexandru Calotoiu, and Torsten Hoefler. Scaling Laws of Global Weather Models, June 2026. URL http://arxiv.org/abs/2602.22962. arXiv:2602.22962 [cs.LG].

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824, Paris, France, October 2023. IEEE. ISBN 979-8-3503-0718-4. doi: 10.1109/ICCV51070.2023.00355. URL https://ieeexplore.ieee.org/document/10377881/.

APPENDIX CONTENTS

A Methodological details **18**

A.1 Data . . . 18
A.1.1 Datasets . . . 18
A.1.2 List of variables . . . 18
A.2 Aurora: the foundation model prior . . . 18
A.3 Xaurora . . . 18
A.3.1 Architecture details . . . 18
A.3.2 Normalisation and network preconditioning . . . 20
A.4 Evaluation . . . 20
A.4.1 Metrics . . . 20
A.4.2 Statistical significance for ablation experiments . . . 21
A.4.3 Forecast settings . . . 22
A.4.4 Baseline models . . . 22
A.4.5 Matched benchmark for Xaurora models . . . 23
A.5 Drift parameterisations . . . 23
A.5.1 Loss function equivalences . . . 23
A.5.2 Mathematical generalisation of the 2D toy example . . . 23
A.6 Denoising Stochastic Interpolants . . . 24
A.6.1 General algorithms . . . 24
A.6.2 Single-backpropagation CRPS . . . 25
A.6.3 Probabilistic SDE-rollout replay buffer training algorithm . . . 26
A.6.4 DSI with pretrained prior requires $\dot{\beta}_0 \neq \tau$ . . . 26
A.7 Training details . . . 27
A.8 Computational requirements . . . 27

B Supplementary results **29**

B.1 Spectral calibration interventions . . . 29
B.1.1 No intervention: White noise . . . 29
B.1.2 Variance scaling . . . 30
B.1.3 Noise colouring . . . 30
B.1.4 Isotropic spherical white noise . . . 31
B.1.5 Noise model fitting with Matérn covariance . . . 31
B.2 Noise floor investigation . . . 33
B.2.1 Generative model and solver configuration . . . 33
B.2.2 Sufficient denoiser training . . . 36
B.2.3 Contribution of the pretrained prior . . . 36
B.2.4 Contribution of the finetuning method . . . 38

B.3 Ablations 38
B.3.1 Main paper ablations details 38
B.3.2 In-depth comparison: Xaurora vs Single step + MAE 39
B.3.3 Ensemble size 39
B.3.4 Finetuning vs training from scratch 40
B.3.5 Naive probabilistic Aurora baseline: MC-DropPath 40
B.4 Additional metrics 40
B.5 Additional comparison with Aurora models 49
B.5.1 Aurora scorecards 49
B.5.2 Forecast visualisation 50

Table 2: **List of variables**. Atmospheric variables are measured on discrete pressure levels (50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, 1000 hPa).

Symbol	Units	Name	Kind
2T	K	2m-temperature	Surface
MSL	Pa	Mean sea level pressure	Surface
10U	$\mathrm{m\,s^{-1}}$	10m-zonal wind component (east-west)	Surface
10V	$\mathrm{m\,s^{-1}}$	10m-meridional wind component (north-south)	Surface
T	K	Temperature	Atmosphere
Z	$\mathrm{m^2/s^2}$	Geopotential	Atmosphere
U	$\mathrm{m\,s^{-1}}$	Zonal wind component (east-west)	Atmosphere
V	$\mathrm{m\,s^{-1}}$	Meridional wind component (north-south)	Atmosphere
Q	$\mathrm{kg\,kg^{-1}}$	Specific humidity	Atmosphere

A Methodological details

A.1 Data

A.1.1 Datasets

ERA5. The global climate reanalysis dataset generated by ECMWF (Hersbach et al., 2020) by combining historical observations with physical models. It spans 1940 to the present at hourly resolution on a 0.25° grid, contains 137 pressure levels from the surface to the top of the atmosphere and includes a broad range of atmospheric, surface, and oceanic variables. In this study, we use a subset of ERA5 from the Weatherbench2 project (Rasp et al., 2024) which is downsampled to 6-hourly resolution and has a reduced set of variables and pressure levels[3] that are listed in Table 2.

HRES-T0. The initial conditions used for operational NWP (Rasp et al., 2024). Operational DLWP models are usually finetuned on this dataset, rendering them theoretically ready for real-time forecasting. We use the WeatherBench2 version of the dataset[4].

A.1.2 List of variables

Variables modelled by Xaurora are the same as the Aurora model. They are listed in Table 2.

A.2 Aurora: the foundation model prior

Aurora (Bodnar et al., 2025) is one of the first atmospheric foundation models (along with ClimaX (Nguyen et al., 2023)). It is competitive with SOTA operational NWP and DLWP models, and was shown to generalise to other tasks with limited finetuning. Aurora is a Transformer-based (Vaswani et al., 2017) deterministic autoregressive model that predicts $\boldsymbol{x}^{t+\Delta t}$ from $\boldsymbol{h}^t$. The model has three main components; (1) a Perceiver-based (Jaegle et al., 2021) encoder that projects 3D atmospheric states into a shared latent grid, (2) a Swin Transformer (Liu et al., 2021) backbone operating on the latents, and (3) a Perceiver-based decoder to project the latents back to physical space. Spatiotemporal information (scale, position, absolute time) is vectorised with Fourier expansions and added to the output of the encoder. A Fourier expansion of the lead time is embedded in the Swin Transformer with AdaLN (Perez et al., 2018; Peebles & Xie, 2023). Three versions of the model are publicly available and are listed in Table 3.

A.3 Xaurora

A.3.1 Architecture details

Figure 6 shows the architecture of the Xaurora denoiser with the changes described in Section 4.4. Table 4 shows the parameter count for each addition. Most of the parameters come from the LoRA

[3] Stored on Google cloud at gs://weatherbench2/datasets/era5/1959-2023_01_10-wb13-6h-1440x721.zarr

[4] Stored on Google cloud at gs://weatherbench2/datasets/hres_t0/2016-2022-6h-1440x721.zarr

Table 3: Aurora model variants in Bodnar et al. (2025).

Model	Size	Accuracy	Key characteristics
Aurora-S	113M	Lowest	Fastest and lightest.
Aurora-L	1.3B	High	Pretrained foundation model.
Aurora-O	1.3B	Highest	Aurora-L fine-tuned for operational weather forecasts.

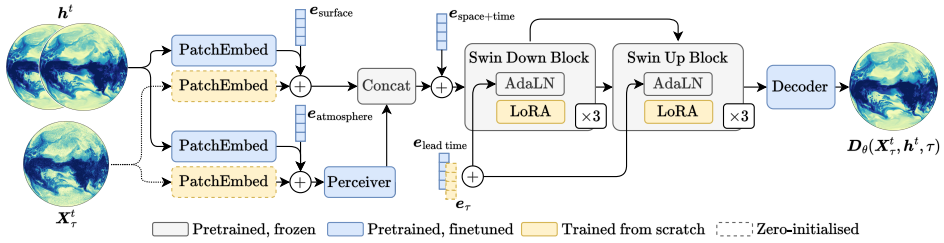

Figure 6: **Xaurora denoiser architecture**. Minimal modifications to Aurora that turn it into the denoiser $\boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau)$. New parameters are initialised such that the model reduces to Aurora before training. Embeddings: $\boldsymbol{e}_{\text{surface}}$, $\boldsymbol{e}_{\text{atmosphere}}$ are Aurora's surface/atmospheric variable embeddings, $\boldsymbol{e}_{\text{space+time}}$ the spatiotemporal position embedding, $\boldsymbol{e}_{\text{lead time}}$ the lead-time embedding, and $\boldsymbol{e}_\tau$ the new (zero-initialised) pseudo-time embedding.

Table 4: Parameter increase in the denoiser compared to Aurora-S and Aurora-L.

Component	Aurora-S	Aurora-L
LoRA	540,672	2,850,816
SDE state $\boldsymbol{X}_\tau^t$	49,664	99,328
Pseudo-time τ	131,584	525,312
Total	721,920	3,475,456
Increase	+0.64%	+0.28%

layers in the Swin transformer. The total parameter increase is less than 1% for Aurora-S and Aurora-L.

The Fourier expansion used to embed the pseudo time τ into a vector of size D is taken from Aurora's encoding of position, scale, level, and time. It can be expressed as follows:

$$\mathrm{Vec}(\tau) = \left[\cos\frac{2\pi\tau}{\lambda_i}, \sin\frac{2\pi\tau}{\lambda_i}\right], \quad 0 \le i < D/2, \tag{10}$$

where

$$\lambda_i = \exp\left(\log\lambda_{\min} + i \cdot \frac{\log\lambda_{\max} - \log\lambda_{\min}}{D/2 - 1}\right), \tag{11}$$

with $\lambda_{\min} = 10^{-6}$ and $\lambda_{\max} = 1$.

A.3.2 Normalisation and network preconditioning

We reuse the per-variable normalisation statistics from Aurora (see Bodnar et al. (2025), Section B.5). In addition, we ensure that the new inputs to the denoiser have a zero-mean and a unit-variance on average by adapting Karras et al. (2022) to the DSI parameterisation:

$$\boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{x}^{t-\Delta t:t}, \tau) \coloneqq c_{\text{out}}(\tau)\tilde{\boldsymbol{D}}_\theta\big(c_{\boldsymbol{X}}(\tau)\boldsymbol{X}_\tau^t, c_{\boldsymbol{x}}(\tau)\boldsymbol{x}^{t-\Delta t:t}, c_\tau(\tau)\,\tau\big). \tag{12}$$

As the physical states $\boldsymbol{x}^{t-\Delta t:t+\Delta t}$ are standardised, $c_{\text{out}}(\tau) = c_{\boldsymbol{x}}(\tau) = 1$. The pseudo-time is vectorised with a Fourier expansion, which is standardised by construction. The SDE state, $\boldsymbol{X}_\tau^t$, has zero mean but non-unit variance, therefore we set the conditioning function to the approximate inverse standard deviation. For simplicity, we ignore the correlation between $\boldsymbol{x}^t$ and $\boldsymbol{x}^{t+\Delta t}$.

$$\boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau) \coloneqq \tilde{\boldsymbol{D}}_\theta\big(c(\tau)\,\boldsymbol{X}_\tau^t,\ \boldsymbol{h}^t,\ \tau\big), \qquad c(\tau) \coloneqq \frac{1}{\sqrt{\mathrm{Var}[\boldsymbol{I}_\tau^t]}} \approx \frac{1}{\sqrt{\alpha_\tau^2 + \beta_\tau^2 + \tau\gamma_\tau^2}}. \tag{13}$$

A.4 Evaluation

A.4.1 Metrics

We use the latest version of the WeatherBench evaluation[5]. All the metrics below are adapted from Rasp et al. (2024). They are computed per variable and per lead time, so we drop both indices: $\boldsymbol{x}$ is the ground truth, and $\hat{\boldsymbol{x}}^{(m)}$ the m-th of the M ensemble members, all on the $N_{\text{lat}} \times N_{\text{lon}}$ grid. The metrics are latitude-weighted to compensate for the non-uniformity of the grid. With $\mathrm{latitude}(i)$ mapping a cell index to its latitude in radians, the weights are

$$w(i) \coloneqq \frac{\cos(\mathrm{latitude}(i))}{\frac{1}{N_{\text{lat}}}\sum_{j=1}^{N_{\text{lat}}}\cos(\mathrm{latitude}(j))}. \tag{14}$$

We denote the number of forecast initialisations with N_{init}, the ensemble mean with $\bar{\boldsymbol{x}} = \frac{1}{M}\sum_m \hat{\boldsymbol{x}}^{(m)}$ and the unbiased sample variance with $\boldsymbol{\sigma}^2 = \frac{1}{M-1}\sum_m(\hat{\boldsymbol{x}}^{(m)} - \bar{\boldsymbol{x}})^2$.

Root Mean Squared Error (RMSE). The RMSE quantifies the discrepancies between predictions and ground truth.

$$\mathrm{RMSE} \coloneqq \sqrt{\frac{1}{N_{\text{init}}N_{\text{lat}}N_{\text{lon}}}\sum_{t=1}^{N_{\text{init}}}\sum_{i=1}^{N_{\text{lat}}}\sum_{j=1}^{N_{\text{lon}}} w(i)\big(\hat{\boldsymbol{x}}_{t,i,j} - \boldsymbol{x}_{t,i,j}\big)^2}. \tag{15}$$

The RMSE can be used to assess an ensemble mean, but is biased by the ensemble size M. It is the right choice to evaluate the effect of different ensemble sizes, but not to compare ensembles of varying sizes.

[5] Adapted from https://github.com/google-research/weatherbenchX/blob/main/public_benchmark/run_benchmark_evaluation.py

Unbiased Ensemble Mean RMSE (uRMSE). The uRMSE removes the finite-ensemble bias $\boldsymbol{\sigma}^2_{t,i,j}/M$ from the residuals, so that ensembles of different sizes can be directly compared.

$$\text{uRMSE} \coloneqq \sqrt{\frac{1}{N_{\text{init}} N_{\text{lat}} N_{\text{lon}}} \sum_{t=1}^{N_{\text{init}}} \sum_{i=1}^{N_{\text{lat}}} \sum_{j=1}^{N_{\text{lon}}} w(i) \left[\left(\bar{\boldsymbol{x}}_{t,i,j} - \boldsymbol{x}_{t,i,j} \right)^2 - \frac{\boldsymbol{\sigma}^2_{t,i,j}}{M} \right]}. \tag{16}$$

Note that uRMSE is not defined for deterministic models since the bias term is singular with $M = 0$.

Fair Continuous Ranked Probability Score (CRPS). The CRPS measures the difference between the cumulative distributions of the forecasts and the predictions (Gneiting & Raftery, 2007). It is defined as the combination of a skill and a spread component. With

$$\text{CRPS}_{\text{skill}} \coloneqq \frac{1}{N_{\text{init}} M N_{\text{lat}} N_{\text{lon}}} \sum_{t=1}^{N_{\text{init}}} \sum_{m=1}^{M} \sum_{i=1}^{N_{\text{lat}}} \sum_{j=1}^{N_{\text{lon}}} w(i) \left| \hat{\boldsymbol{x}}^{(m)}_{t,i,j} - \boldsymbol{x}_{t,i,j} \right| \tag{17}$$

$$\text{CRPS}_{\text{spread}} \coloneqq \frac{1}{N_{\text{init}}} \sum_{t=1}^{N_{\text{init}}} \frac{1}{2M(M-1)} \sum_{m=1}^{M} \sum_{\bar{m}=1}^{M} \frac{1}{N_{\text{lat}} N_{\text{lon}}} \sum_{i=1}^{N_{\text{lat}}} \sum_{j=1}^{N_{\text{lon}}} w(i) \left| \hat{\boldsymbol{x}}^{(m)}_{t,i,j} - \hat{\boldsymbol{x}}^{(\bar{m})}_{t,i,j} \right|, \tag{18}$$

the *fair* CRPS, which is unbiased for the $M \to \infty$ limit and therefore comparable across ensemble sizes, is given by

$$\text{CRPS} \coloneqq \text{CRPS}_{\text{skill}} - \text{CRPS}_{\text{spread}}. \tag{19}$$

Spread-Skill Ratio (SSR). The SSR measures the calibration as the ratio of the ensemble spread and the unbiased RMSE of the ensemble mean.

$$\text{SSR} \coloneqq \frac{\sqrt{\frac{1}{N_{\text{init}} N_{\text{lat}} N_{\text{lon}}} \sum_{t=1}^{N_{\text{init}}} \sum_{i=1}^{N_{\text{lat}}} \sum_{j=1}^{N_{\text{lon}}} w(i)\, \boldsymbol{\sigma}^2_{t,i,j}}}{\text{uRMSE}}. \tag{20}$$

Power Spectral Density (PSD). We use the zonal energy spectrum of Rasp et al. (2024). Along the circle of constant latitude φ, which has $L = N_{\text{lon}}$ samples $f_0(\varphi), \ldots, f_{L-1}(\varphi)$ and circumference $C(\varphi) = 2\pi R_e \cos\varphi$, the discrete Fourier transform is $F_k(\varphi) = \frac{1}{L} \sum_{l=0}^{L-1} f_l(\varphi)\, e^{-i2\pi kl/L}$, and the spectrum of that single circle is

$$S_0(\varphi) \coloneqq C(\varphi)\, |F_0(\varphi)|^2, \qquad S_k(\varphi) \coloneqq 2\, C(\varphi)\, |F_k(\varphi)|^2, \quad k = 1, \ldots, \lfloor L/2 \rfloor, \tag{21}$$

where the factor 2 for $k > 0$ accounts for both positive and negative frequency content. This normalisation makes Parseval's relation hold at each latitude, $\int_0^{C(\varphi)} |f(l,\varphi)|^2 \, \mathrm{d}l \approx \frac{C(\varphi)}{L} \sum_l |f_l(\varphi)|^2 = \sum_k S_k(\varphi)$, so $S_k(\varphi)$ decomposes that circle's total power by wavenumber k. The reported power spectral density is the latitude average

$$P(k) \coloneqq \frac{\sum_{\varphi \in \Phi} w(\varphi)\, S_k(\varphi)}{\sum_{\varphi \in \Phi} w(\varphi)}, \tag{22}$$

over the latitude set Φ with weights $w(\varphi)$. Unless stated otherwise, we use the entire globe ($\Phi = [0°, 90°]$) in our evaluation.

A.4.2 Statistical significance for ablation experiments

In the ablation experiments, we provide statistical significance. We obtain the intervals by re-sampling the metrics over our 56 initialisations before temporal aggregation. Because we are interested in the significance of the comparison between two models, we use paired bootstrapping by comparing the models on the same set of resampled initialisations with replacement. This approach removes the weather variability that dominates with 56 initialisations. Formally, for a metric $\mathcal{M}$, we bootstrap the relative difference $\Delta = (\mathcal{M}_{\text{model}} - \mathcal{M}_{\text{ref}})/\mathcal{M}_{\text{ref}}$ with 10,000 replicates and report percentile intervals and two-sided percentile p-values at $\alpha = 0.05$.

The reported significance does not depend on this choice of inference procedure. Repeating the analysis with a stationary bootstrap and with paired t-tests on the 56 per-initialisation score (as used by Kossaifi et al. (2026)) shifts the interval bounds slightly, but leaves every significance label in Table 1 unchanged.

Note that we downloaded the metrics of the baseline models from WeatherBench and did not run the evaluation ourselves. Therefore, no statistical significance is reported when comparing Xaurora with these models (e.g. Figure 4), which is the norm in the development of DLWP.

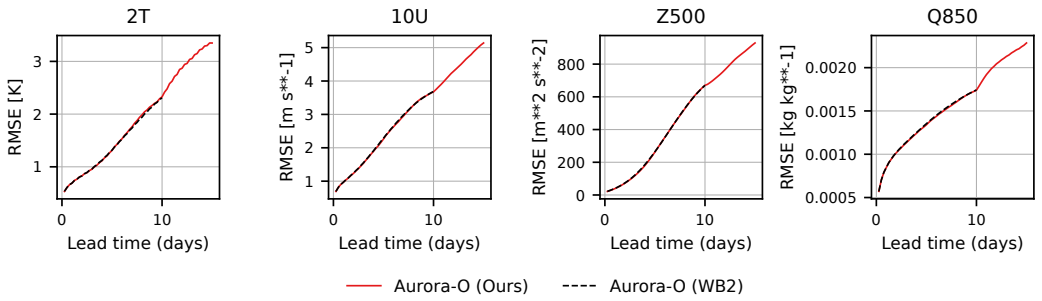

Figure 7: **Aurora-O validation.** RMSE of Aurora-O from the official WeatherBench2 (WB2, dashed black) and from our own run (Ours, red). The two runs are essentially equivalent, with negligible variations coming from different sets of initialisations. The WB2 run only goes up to 10 days and hides the shut down of the rollout training mechanism used in Aurora-O (discontuity at 10 days).

A.4.3 Forecast settings

All forecasts are initialised on 56 different evenly spread dates across 2022 (either at 12 or 00 UTC). This is in line with Kossaifi et al. (2026), who report statistically significant results using only 28 initialisations, massively decreasing the compute and storage requirements over the WeatherBench approach (2 initialisations per day across 2022). To reduce the forecasts size on disc and maximise write speed, we store all surface variables, but only a subset of pressure levels (250, 500, 700, 850 hPa).

Xaurora. All Xaurora forecasts use Algorithm 2 with a Heun solver and 10 integration steps (*i.e.* 20 neural network evaluations per SDE). We generate ensemble forecasts of 10 members to minimise computational cost without compromising on the metrics.

Aurora. Aurora-O metrics are only publicly available up to 10 days of lead time[6] and the metrics of Aurora-S and Aurora-L are not available. Therefore, we run all the Aurora forecasts discussed in this paper ourselves according to Appendix A.4.3. Figure 7 shows that our results align with the official evaluation.

A.4.4 Baseline models

We compare the predictions with the following baselines:

- **Operational ensemble models:** the Integrated Forecasting System Ensemble (IFS ENS)[7] and GenCast operational (GenCast-O)[8] (Alet et al., 2025);
- **Operational deterministic models:** Aurora operational (Aurora-O) (Bodnar et al., 2025);
- **non-operational Aurora versions:** Aurora-Large (L) and Aurora-S (S) (Bodnar et al., 2025).

Table 3 lists the characteristics of the different Aurora models. We do not use Aurora 1.5 (Weyn et al., 2026), a recent probabilistic version of Aurora-O trained with CRPS, as at the time of writing it is only available as an unpublished manuscript.

Operational systems are evaluated against HRES-T0, others against ERA5. Therefore, the models are not strictly comparable. Nevertheless, this is standard practice in the field, since running forecasts on different initial conditions and evaluating them on different analyses is prohibitively expensive and potentially also unfair.

[6] https://storage.googleapis.com/weatherbench2/benchmark_results/aurora_vs_hres_t0_1440x721_2022.nc

[7] https://storage.googleapis.com/weatherbench2/benchmark_results/ens_mean_vs_hres_t0_1440x721_2022.nc

[8] https://storage.googleapis.com/weatherbench2/benchmark_results/gencast_operational_100m_uv_vs_hres_t0_1440x721_2022.nc

Table 5: Neural network, target and drift term for each parameterisation

	Network	**Target**	**Drift** ($\mathbb{E}[\boldsymbol{R}_\tau^t \mid \boldsymbol{X}_\tau^t, \boldsymbol{h}^t]$)
$\boldsymbol{B}$-prediction	$\boldsymbol{B}_\theta$	$\boldsymbol{R}_\tau^t$	$\boldsymbol{B}_\theta$
$\boldsymbol{W}$-prediction	$\boldsymbol{W}_\theta$	$\boldsymbol{W}_\tau$	$\dot{\alpha}_\tau \boldsymbol{x}^t + \frac{\dot{\beta}_\tau}{\beta_\tau}\left[\boldsymbol{X}_\tau^t - \alpha_\tau \boldsymbol{x}^t - \gamma_\tau \boldsymbol{W}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau)\right] + \dot{\gamma}_\tau \boldsymbol{W}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau)$
$\boldsymbol{x}_1$-prediction	$\boldsymbol{D}_\theta$	$\boldsymbol{x}^{t+\Delta t}$	$\dot{\alpha}_\tau \boldsymbol{x}^t + \dot{\beta}_\tau \boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau) + \frac{\dot{\gamma}_\tau}{\gamma_\tau}\left[\boldsymbol{X}_\tau^t - \alpha_\tau \boldsymbol{x}^t - \beta_\tau \boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau)\right]$

A.4.5 Matched benchmark for Xaurora models

In some situations, when comparing different configurations of the same Xaurora checkpoint, we run paired evaluations that remove the variability of the weather systems and the Wiener process by using a common set of initialisations (usually smaller than the 56 used in Appendix A.4.3) and the same noise samples (fixed seed). This allows faster and more accurate benchmarks. Note that this is not used in the main paper ablation experiments. This procedure is also used to validate the models during training.

A.5 Drift parameterisations

Table 5 lists the drift parameterisations of the Stochastic Interpolant discussed in Section 4.1. We obtain these results by developing the drift as an expectation and isolating relevant terms in Equation (1),

$$\boldsymbol{B}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau) = \mathbb{E}[\boldsymbol{R}_\tau^t \mid \boldsymbol{X}_\tau^t, \boldsymbol{h}^t] = \dot{\alpha}_\tau \boldsymbol{x}^t + \dot{\beta}_\tau \mathbb{E}[\boldsymbol{x}^{t+\Delta t} \mid \boldsymbol{X}_\tau^t, \boldsymbol{h}^t] + \dot{\gamma}_\tau \mathbb{E}[\boldsymbol{W}_\tau \mid \boldsymbol{X}_\tau^t, \boldsymbol{h}^t]. \quad (23)$$

A.5.1 Loss function equivalences

Similarly to Li & He (2026), the $\boldsymbol{x}_1$-space loss in Equation (7) can be expressed in $\boldsymbol{B}$- and $\boldsymbol{W}$-space by a single τ-dependent reweighing:

$$\mathcal{L}_{\boldsymbol{B}|\boldsymbol{D}}(\theta) = \int_0^1 \mathbb{E}_{(\boldsymbol{h}^t, \boldsymbol{x}^{t+\Delta t}) \sim p_{\text{data}}, \boldsymbol{W}_\tau \sim \mathcal{N}(\boldsymbol{0}, \tau \mathbf{I})} \Big[\underbrace{\left(\dot{\beta}_\tau - \frac{\dot{\gamma}_\tau}{\gamma_\tau}\beta_\tau\right)^2}_{=:\lambda_{\boldsymbol{B}|\boldsymbol{D}}(\tau)} \left\| \boldsymbol{D}_\theta(\boldsymbol{I}_\tau^t, \boldsymbol{h}^t, \tau) - \boldsymbol{x}^{t+\Delta t} \right\|_2^2 \Big] \mathrm{d}\tau \quad (24)$$

$$\mathcal{L}_{\boldsymbol{W}|\boldsymbol{D}}(\theta) = \int_0^1 \mathbb{E}_{(\boldsymbol{h}^t, \boldsymbol{x}^{t+\Delta t}) \sim p_{\text{data}}, \boldsymbol{W}_\tau \sim \mathcal{N}(\boldsymbol{0}, \tau \mathbf{I})} \Big[\underbrace{\left(\frac{\beta_\tau}{\gamma_\tau}\right)^2}_{=:\lambda_{\boldsymbol{W}|\boldsymbol{D}}(\tau)} \left\| \boldsymbol{D}_\theta(\boldsymbol{I}_\tau^t, \boldsymbol{h}^t, \tau) - \boldsymbol{x}^{t+\Delta t} \right\|_2^2 \Big] \mathrm{d}\tau. \quad (25)$$

The derivation is purely algebraic. For $\lambda_{\boldsymbol{B}|\boldsymbol{D}}(\tau)$, we plug Equation (6) and the definition of $\boldsymbol{R}_\tau^t$ in Equation (3). For $\lambda_{\boldsymbol{W}|\boldsymbol{D}}(\tau)$, use $\boldsymbol{W}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau) = \frac{1}{\gamma_\tau}[\boldsymbol{X}_\tau^t - \alpha_\tau \boldsymbol{x}^t - \beta_\tau \boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau)]$ and $\mathcal{L}_{\boldsymbol{W}}(\theta) \coloneqq \int_0^1 \mathbb{E}_{(\boldsymbol{h}^t, \boldsymbol{x}^{t+\Delta t}) \sim p_{\text{data}}, \boldsymbol{W}_\tau \sim \mathcal{N}(\boldsymbol{0}, \tau \mathbf{I})} \left[\left\| \boldsymbol{W}_\theta(\boldsymbol{I}_\tau^t, \boldsymbol{h}^t, \tau) - \boldsymbol{W}_\tau \right\|_2^2 \right] \mathrm{d}\tau$. Note that we train in $\boldsymbol{x}_1$-space and leave the comparison for future work.

A.5.2 Mathematical generalisation of the 2D toy example

The 2-dimensional toy example shown in Figure 2 supports that DSI parameterisation is preferable when finetuning an ideal prior model of the form $\boldsymbol{G} = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0]$ (we ignore the temporal dimension t for conciseness, without loss of generality). This section provides the same motivation from a mathematical perspective valid for problems of arbitrary finite dimension.

The central idea is always the same: we use the prior $\boldsymbol{G}$ in lieu of the network in the expression of the drift corresponding to each parameterisation (last column of Table 5). The obtained drift, referred to as $\boldsymbol{B}(\boldsymbol{G})$ below, is then substituted in the SI SDE (Equation (4)). We can then approximate the mean SDE trajectory (the bold lines in Figure 2) by ignoring the Wiener process and solving the following ODE

$$\mathrm{d}\bar{\boldsymbol{X}}_\tau = \boldsymbol{B}(\boldsymbol{G})\mathrm{d}\tau, \quad \bar{\boldsymbol{X}}_0 = \boldsymbol{x}_0. \quad (26)$$

We use $\alpha_\tau = \gamma_\tau = 1 - \tau$, $\beta_\tau = \tau$, and $\boldsymbol{x}^t = \boldsymbol{x}_0$.

$\boldsymbol{B}$-prediction. Let $\boldsymbol{B} = \boldsymbol{G}$:

$$\boldsymbol{B}(\boldsymbol{G}) = \boldsymbol{G} \implies \mathrm{d}\bar{\boldsymbol{X}}_\tau = \boldsymbol{B}(\boldsymbol{G})\mathrm{d}\tau = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0]\mathrm{d}\tau.$$

Hence, $\bar{\boldsymbol{X}}_\tau = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] \cdot \tau + \boldsymbol{x}_0$. The ODE endpoint is therefore $\bar{\boldsymbol{X}}_1 = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] + \boldsymbol{x}_0 \neq \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0]$. Hence, $\boldsymbol{B}$-prediction diverges.

$\boldsymbol{W}$-prediction. Let $\boldsymbol{W} = \boldsymbol{G}$:

$$\begin{aligned}
\boldsymbol{B}(\boldsymbol{G}) &= \dot{\alpha}_\tau \boldsymbol{x}_0 + \frac{\dot{\beta}_\tau}{\beta_\tau}\big[\boldsymbol{X}_\tau - \alpha_\tau \boldsymbol{x}_0 - \gamma_\tau \boldsymbol{G}\big] + \dot{\gamma}_\tau \boldsymbol{G} \\
&= -\boldsymbol{x}_0 + \frac{1}{\tau}\big[\boldsymbol{X}_\tau - (1-\tau)\boldsymbol{x}_0 - (1-\tau)\boldsymbol{G}\big] - \boldsymbol{G} \\
&= \frac{\boldsymbol{X}_\tau - (\boldsymbol{x}_0 + \boldsymbol{G})}{\tau}.
\end{aligned}$$

As mentioned in the main text, we clip the denominator of the second term to 0.05 to avoid numerical instabilities. Hence,

$$\mathrm{d}\bar{\boldsymbol{X}}_\tau = \frac{\bar{\boldsymbol{X}}_\tau - (\boldsymbol{x}_0 + \boldsymbol{G})}{\max(\tau, 0.05)}\mathrm{d}\tau = \frac{\bar{\boldsymbol{X}}_\tau - (\boldsymbol{x}_0 + \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0])}{\max(\tau, 0.05)}\mathrm{d}\tau.$$

Solving for $0 \leq \tau < 0.05$ and $0.05 \leq \tau \leq 1$ separately and imposing continuity at $\tau = 0.05$ gives

$$\bar{\boldsymbol{X}}_\tau = \begin{cases} \boldsymbol{x}_0 + \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] - \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] e^{\tau/0.05} & \text{if } 0 \leq \tau < 0.05, \\ \boldsymbol{x}_0 + \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] - \frac{e}{0.05}\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] \cdot \tau & \text{if } 0.05 \leq \tau \leq 1. \end{cases}$$

In particular, $\bar{\boldsymbol{X}}_1 = \boldsymbol{x}_0 + \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] - \frac{e}{0.05}\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] \neq \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0]$. Hence, $\boldsymbol{W}$-prediction diverges.

$\boldsymbol{x}_1$-prediction (DSI). Let $\boldsymbol{D} = \boldsymbol{G}$:

$$\begin{aligned}
\boldsymbol{B}(\boldsymbol{G}) &= \dot{\alpha}_\tau \boldsymbol{x}_0 + \dot{\beta}_\tau \boldsymbol{G} + \frac{\dot{\gamma}_\tau}{\gamma_\tau}\big[\boldsymbol{X}_\tau - \alpha_\tau \boldsymbol{x}_0 - \beta_\tau \boldsymbol{G}\big] \\
&= -\boldsymbol{x}_0 + \boldsymbol{G} - \frac{1}{1-\tau}\big[\boldsymbol{X}_\tau - (1-\tau)\boldsymbol{x}_0 - \tau \boldsymbol{G}\big] \\
&= -\frac{1}{1-\tau}(\bar{\boldsymbol{X}}_\tau - \boldsymbol{G}).
\end{aligned}$$

Hence,

$$\mathrm{d}\bar{\boldsymbol{X}}_\tau = -\frac{1}{1-\tau}(\bar{\boldsymbol{X}}_\tau - \boldsymbol{G})\mathrm{d}\tau = -\frac{1}{1-\tau}(\bar{\boldsymbol{X}}_\tau - \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0])\mathrm{d}\tau \implies \bar{\boldsymbol{X}}_\tau = \boldsymbol{x}_0 + (\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0] - \boldsymbol{x}_0) \cdot \tau,$$

which converges to the correct endpoint, $\bar{\boldsymbol{X}}_1 = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0]$.

The residual prediction case. Aurora is parameterised for direct prediction. Another common approach in autoregressive forecasting is *residual prediction*, *i.e.* $\boldsymbol{G} = \mathbb{E}[\boldsymbol{x}_1 - \boldsymbol{x}_0 \mid \boldsymbol{x}_0]$. In that case, the preferable parameterisation would be $\boldsymbol{B}$-prediction. Proceeding as above with the new prior,

$$\boldsymbol{B}(\boldsymbol{G}) = \boldsymbol{G} \implies \mathrm{d}\bar{\boldsymbol{X}}_\tau = \boldsymbol{G}\mathrm{d}\tau = \mathbb{E}[\boldsymbol{x}_1 - \boldsymbol{x}_0 \mid \boldsymbol{x}_0]\mathrm{d}\tau,$$

and $\bar{\boldsymbol{X}}_\tau = \mathbb{E}[\boldsymbol{x}_1 - \boldsymbol{x}_0 \mid \boldsymbol{x}_0] \cdot \tau + \boldsymbol{x}_0$, which has the correct endpoint, $\bar{\boldsymbol{X}}_1 = \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_0]$. Nevertheless, the network still targets $\boldsymbol{R}_\tau^t$ and could therefore be less robust to the curse of dimensionality, following the arguments of Li & He (2026).

A.6 Denoising Stochastic Interpolants

A.6.1 General algorithms

Algorithm 1 and Algorithm 2 show the DSI training and sampling algorithms. Our probabilistic SDE-rollout training algorithm is defined in Appendix A.6.3.

Algorithm 1 Vanilla autoregressive DSI training

Input: $(\alpha_\tau, \beta_\tau, \gamma_\tau)$, training data $\mathcal{D}_{\text{train}}$, pretrained $\boldsymbol{D}_\theta$, number of training samples N_{samples}, optimiser
for $n = 1$ **to** N_{samples} **do**
  $(\boldsymbol{x}^{t-\Delta t}, \boldsymbol{x}^t, \boldsymbol{x}^{t+\Delta t}) \sim \mathcal{D}$
  $\tau \sim U[0, 1]$
  $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\boldsymbol{W}_\tau \leftarrow \sqrt{\tau}\boldsymbol{z}$
  $\boldsymbol{I}_\tau^t \leftarrow \alpha_\tau \boldsymbol{x}^t + \beta_\tau \boldsymbol{x}^{t+\Delta t} + \gamma_\tau \boldsymbol{W}_\tau$ (this Vanilla algorithm explicitly forms the interpolant.)
  $\ell(\theta) \leftarrow \left\| \boldsymbol{D}_\theta(\boldsymbol{I}_\tau^t, \boldsymbol{h}^t, \tau) - \boldsymbol{x}^{t+\Delta t} \right\|_2^2$ (from Equation (7))
  $\theta \leftarrow \text{optimiser}\big(\ell(\theta), \theta\big)$
end for
Output: Finetuned denoiser $\boldsymbol{D}_\theta$

Algorithm 2 Autoregressive DSI sampling

Input: Initial conditions $\boldsymbol{h}^t$, trained denoiser $\boldsymbol{D}_\theta$, SDE steps S, forecast steps K, SDE Solver
$\Delta\tau \leftarrow 1/S$
$\hat{\boldsymbol{h}}^t \leftarrow \boldsymbol{h}^t$ (running history; after the first step it holds generated states)
for $k = 0$ **to** $K - 1$ **do**
  $s \leftarrow t + k\Delta t$
  $\tau \leftarrow 0$
  $\boldsymbol{X}_0^s \leftarrow$ last element of $\hat{\boldsymbol{h}}^s$
  for $n = 1$ **to** S **do**
    $\tau' \leftarrow \tau + \Delta\tau$
    $\boldsymbol{X}_{\tau'}^s \leftarrow \text{SDE Solver}\big(\boldsymbol{X}_\tau^s,\ \boldsymbol{B}_\theta(\cdot, \hat{\boldsymbol{h}}^s, \cdot),\ \tau,\ \Delta\tau\big)$ (solver needs the drift function)
    $\tau \leftarrow \tau'$
  end for
  $\hat{\boldsymbol{x}}^{s+\Delta t} \leftarrow \boldsymbol{X}_1^s$
  $\hat{\boldsymbol{h}}^{s+\Delta t} \leftarrow (\boldsymbol{X}_0^s, \hat{\boldsymbol{x}}^{s+\Delta t})$
end for
Output: forecast $\left\{\hat{\boldsymbol{x}}^{t+k\Delta t}\right\}_{k=1}^K$

A.6.2 SINGLE-BACKPROPAGATION CRPS

The fair CRPS loss used to train Xaurora is shown in the main paper, in Equation (9). We use the same variable weights (ω) as Aurora[9], such that at $\tau = 0$ where the spread term ($\frac{1}{2}\left\|\hat{\boldsymbol{x}}^{(m)} - \hat{\boldsymbol{x}}^{(\bar{m})}\right\|_\omega$) is zero, the loss is equivalent to Aurora's weighted MAE. Both members are functions of θ, so by the chain rule

$$\nabla_\theta \, \ell_{\text{fCRPS}} = \partial_{\hat{\boldsymbol{x}}^m} \ell \cdot \partial_\theta \hat{\boldsymbol{x}}^{(m)} + \underbrace{\partial_{\hat{\boldsymbol{x}}^{\bar{m}}} \ell \cdot \partial_\theta \hat{\boldsymbol{x}}^{(\bar{m})}}_{=0}. \tag{27}$$

To avoid backpropagating through both trajectories, we supervise only the $\hat{\boldsymbol{x}}^{(m)}$-trajectory and apply a stop-gradient to $\hat{\boldsymbol{x}}^{(\bar{m})}$. Hence $\partial_\theta \hat{\boldsymbol{x}}^{(\bar{m})} = 0$ and the second term vanishes.

The roles $\bar{m}$ and m are assigned at random between the two members. Since ℓ_{fCRPS} is symmetric in its two member arguments and the members are exchangeable draws from the forecast distribution, the two terms have equal expectation. The one-sided estimator is therefore an unbiased estimate of $\frac{1}{2}\nabla_\theta \, \ell_{\text{fCRPS}}$. The constant $1/2$ scaling is compensated by the learning rate and the the stationary points of ℓ_{fCRPS} remain unchanged. We use a replay buffer to emulate rollouts, which introduces a harmless off-policy relative to teacher forcing (see Figures 8 and 9).

[9] Surface variables are weighted by 0.25 and atmospheric ones by 1. Individual variables also have dedicated weights, 1.5 for MSL, 0.77 for 10U, 0.66 for 10V, 3.0 for 2T, 2.8 for Z, 0.8 for Q, 1.7 for T, 0.87 for U, 0.6 for V. No latitude-weighting is used.

A.6.3 PROBABILISTIC SDE-ROLLOUT REPLAY BUFFER TRAINING ALGORITHM

The algorithm used to train Xaurora on ensemble forecasts is shown in Algorithm 3. Below, we summarise the main characteristics of the algorithm.

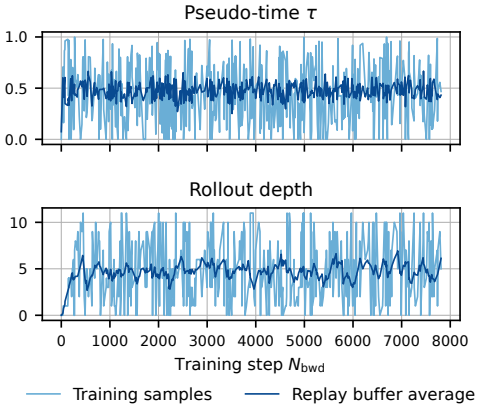

Figure 8: **Replay-buffer rollout training statistics. Top:** Time series of the pseudo-time (τ) of the samples **Bottom:** Time series of the rollout depth of the samples. The values seen by the model at each training steps are in light blue and the replay buffer averages are depicted in dark blue. The x-axis shows the training step N_{bwd} as defined in Table 6. These plots show that throughout the training, the denoiser is exposed to a variety of pseudo-times (mean=0.5) and rollout steps (mean=4.7).

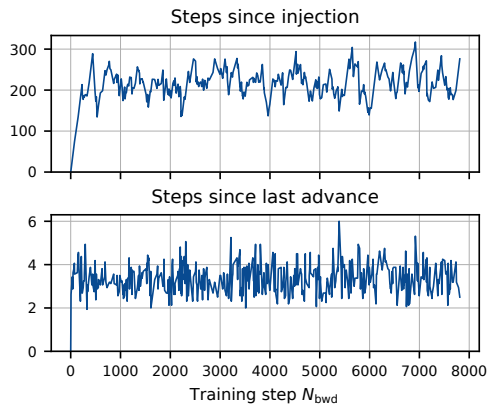

Figure 9: **Replay-buffer training off-policy discrepancy. Top:** Buffer mean sample age, *i.e.*, number of training steps since the trajectory was first pushed to the buffer. **Bottom:** Average number of training steps since a sample was updated, *i.e.*, popped, advanced by one SDE step, and pushed back to the buffer. The x-axis shows the training step N_{bwd} as defined in Table 6. These plots characterise the minor off-policy discrepancy introduced by the replay buffer compared to a naive rollout training strategy (intractable).

Leak-free trajectories. In the single-step autoregressive DSI training algorithm (Algorithm 1), the target $\boldsymbol{x}^{t+\Delta t}$ can leak into the model input through the interpolant $\boldsymbol{I}_\tau^t$. We start every rollouts at $\tau = 0$, where $\boldsymbol{I}_0^t = \boldsymbol{x}^t$, and evolve SDE trajectories using only the denoiser drift (*i.e.* the network sees $\boldsymbol{X}_\tau^t$, never $\boldsymbol{I}_\tau^t$). This ensures that the target $\boldsymbol{x}^{t+\Delta t}$ appears solely in the loss calculation and that the model trained on the rollout distribution (rather than the distribution of the interpolant).

Replay buffer design. The replay buffer is a FIFO list of SDE states ($\boldsymbol{X}_\tau^t$) with their associated history ($\hat{\boldsymbol{h}}^t$), (pseudo-)temporal coordinates (t, τ) and rollout information (*e.g.* rollout depth). All tensors are kept in memory in FP16.

Hyperparameters. Our replay buffer algorithm has a few hyperparameters that define the rollout training statistics. Their values are listed in Table 6. Figure 8 shows the pseudo-time and rollout depth over the course of the training.

Probabilistic training. We unroll pairs of trajectories (*i.e.* 2 members) that we supervise with the single-backpropagation CRPS described in Equation (9) and Appendix A.6.2.

A.6.4 DSI WITH PRETRAINED PRIOR REQUIRES $\dot{\beta}_0 \neq \tau$

We show that the warm start properties of the DSI require $\dot{\beta}_0 \neq 0$. More specifically, this means that it is preferable to use a linear interpolant ($\beta_\tau = \tau$) rather than a quadratic one ($\beta_\tau = \tau^2$) when finetuning a pretrained prior using the DSI framework.

Substituting $\alpha_\tau = \gamma_\tau = 1 - \tau$ and $\beta_\tau = \tau^2$ in Equation (6), we get:

$$\boldsymbol{B}_\tau^{\text{quadratic}}(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau) = -\frac{1}{1-\tau}\left[\boldsymbol{X}_\tau^t - \tau(2-\tau)\boldsymbol{D}_\theta(\boldsymbol{X}_\tau^t, \boldsymbol{h}^t, \tau)\right]. \tag{28}$$

At $\tau = 0$, $\boldsymbol{B}_0^{\text{quadratic}} = -\boldsymbol{x}^t$, which is independent of the model output and generally does not point in the direction of $\boldsymbol{x}^{t+\Delta t}$. This wrong initial step must be corrected by the subsequent integration steps,

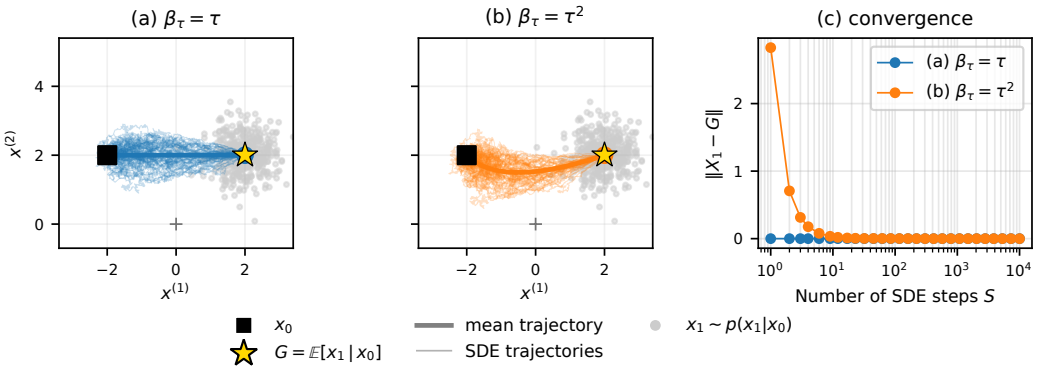

Figure 10: **Denoiser parameterisation: effect of the interpolation schedule β_τ. (a,b)** Warm-start SDE trajectories for the denoiser with a linear schedule $\beta_\tau = \tau$ and a quadratic schedule $\beta_\tau = \tau^2$ (same 2D setup and markers as Fig. 2). **(c)** Distance of the mean SDE trajectory endpoint to the unbiased prior, $\|\boldsymbol{X}_1 - \boldsymbol{G}_\phi\|$, versus the number of integration steps (Euler–Maruyama). The linear schedule is exact regardless of the integration discretisation, whereas the quadratic schedule requires more than 20 steps to converge.

which can be directly observed in panel (c) of Figure 10 (same toy setup as Figure 2); the quadratic formulation does not exactly reach $\boldsymbol{G}_\phi$ for small numbers of SDE steps because it must correct the incorrect initial step. Meanwhile, the linear interpolant is exact regardless of the integration scheme.

Note that any choice of β_τ, such that $\dot{\beta}_0 = 0$ will suffer from an initial drift independent of the model prediction. Conversely, any β_τ with non-zero initial derivative could theoretically succeed, for instance $\beta_\tau = \sin(\pi\tau/2)$.

A.7 Training details

We finetune Aurora-S with the SDE-rollout replay buffer described in Algorithm 3. The denoiser is trained in FP32 with a cosine annealing learning rate, no weight decay, and an AdamW optimiser. The total gradient norm is clipped to 1. Table 6 summarises the hyperparameters. Following the WeatherBench standards, we train Xaurora on ERA5 from 1979 to 2020, validate on 2021, and test on 2022. Aurora-S was pretrained on the same ERA5 splits, so there is no risk of data leakage.

A.8 Computational requirements

We argue that (1) the training strategy of Xaurora is cheap compared to existing probabilistic finetuning methods, and (2) inference is also cheap relative to other generative DLWP models.

Table 7 shows the computational requirements of training and inference for Aurora and Xaurora models. The number of samples and training time required to train convert Aurora-S into Xaurora is significantly lower than the pretraining of Aurora models. Xaurora also converts Aurora-S much faster than Aurora 1.5, which requires 4 times more samples and about 20 times more GPU hours to learn the conditional distribution from Aurora-O. Furthermore, the rollout finetuning that takes Aurora-L to Aurora-O requires an additional 276k samples, which is on the same order of magnitude as the Xaurora finetuning. This allows us to call our finetuning method *cheap*, in relative terms.

Inference is more expensive with generative models than their single-step counterparts, as the former requires multiple function evaluations for a single physical rollout step (nested rollout from Figure 1). Thanks to the warm-start and the DSI, Xaurora can use a small number of integration steps (10) with a Heun solver, keeping the 24 h forecast time under a minute (single member). Note that solving the SDE requires some data manipulations (*e.g.* concatenating tensors, moving data in and out of the GPUs), which adds some computational overhead and explains the non-linear scaling from Aurora-S to Xaurora. The ATLAS paper (Kossaifi et al., 2026) reports that generating a single step forecast[10]

[10] 6 h for ATLAS models, 12 h for GenCast

Table 6: **Xaurora hyperparameters**

Symbol	Meaning	Value
–	Aurora variant	Aurora-S
p_{dropout}	Dropout rate (Hinton et al., 2012)	0.0
p_{droppath}	Stochastic depth (Huang et al., 2016)	0.0
α_τ	Stochastic Interpolant coefficient	$1-\tau$
β_τ	Stochastic Interpolant coefficient	τ (according to Appendix A.6.4)
γ_τ	Stochastic Interpolant coefficient	$1-\tau$
N_{batch}	batch size per rank	1
N_{samples}	total training samples	500k
N_{accum}	gradient accumulation steps	4
N_{GPU}	number of GPUs/ranks	16
N_{bwd}	backward steps per rank ($\frac{N_{\text{samples}}}{N_{\text{batch}} N_{\text{accum}} N_{\text{GPU}}}$)	7812
–	Optimiser	AdamW (Loshchilov & Hutter, 2019)
η_{min}	Initial learning rate	0.0
$\eta_{\text{max}}^{\text{base}}$	peak LR for pre-existing parameters	5×10^{-5}
$\eta_{\text{max}}^{\text{new}}$	peak LR for new parameters	1×10^{-3}
$\eta_{\text{end}}^{\text{base}}$	final LR for pre-existing parameters	5×10^{-5}
$\eta_{\text{end}}^{\text{new}}$	final LR for new parameters	5×10^{-5}
N_{warmup}	warm-up (backward) steps	781 (10%)
N_{B}	replay buffer capacity (per rank)	16
p_{inject}	fresh sample injection rate	5×10^{-3}
K_{max}	maximal rollout depth	12 (=72 h)
S	average SDE steps	10
j	$\Delta\tau$ sampling jitter	0.5

Table 7: **Computational requirements of training and running Aurora and Xaurora models.** Training times only include pretraining for Aurora-S and Aurora-L. GPUs are 80GB A100. Aurora and Aurora 1.5 numbers are taken from (Bodnar et al., 2025; Weyn et al., 2026). Aurora 1.5 numbers indicate additional finetuning over Aurora-O. The last columns shows an untrained Xaurora model initialised with Aurora-L, hence the training numbers are not provided. We also show the number of function evaluation (NFE). Xaurora uses 10 integration steps with a Heun solver (2 NFE per pseudo-step).

	Aurora-L	**Aurora-S**	**Aurora 1.5**	**Xaurora**	**Xaurora init. with Aurora-L**
Training samples	3.8M	3.8M	2.1M	500k	N/A
Training time	2 weeks	unknown	2 weeks	42 hours	N/A
Number of training GPUs	32	unknown	24 to 32	16	N/A
6 h forecast time (in seconds)	2.0	0.6	unknown	13.1	54.6
24 h forecast time (in seconds)	8.2	2.4	unknown	52.3	218.2
NFE per 6 h-step	1	1	1	20	20

on a single A100 GPU takes 140 s for GenCast[11], 94 s for ATLAS-SI, and 88 s for ATLAS-EDM (diffusion). These numbers cannot be directly compared to Table 7 as the evaluation code should be matched exactly. Nevertheless, for the same task, Xaurora takes 13.1 s and 54.6 s when initialised from Aurora-L, which is arguably *cheap* relative to other generative DLWP models.

[11]The GenCast paper claims that a 15-day forecast is generated in 8 minutes on a single Cloud TPUv5 device, which amounts to 16 s for a single timestep (still slower than Xaurora). We use the ATLAS numbers as they are based on the same hardware as ours.

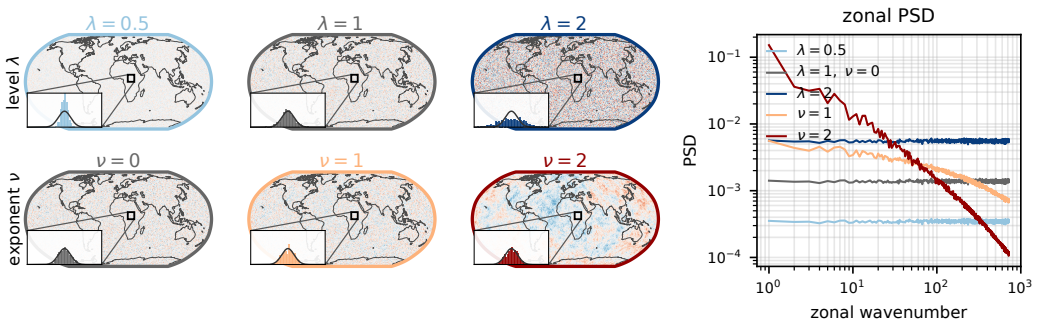

Figure 11: **Noise fields and PSD used in the spectral calibration analysis. Top three:** Fields resulting from scaling the variance of a normal distribution by λ (used Appendix B.1.2). **Bottom three:** fields characterised by a power law PSD, $\mathrm{PSD}(k) \propto k^{-\nu}$ (used in Appendix B.1.3). **Right panel:** PSD of each fields. The inserts in the field plots show the distribution at a specific location over 500 draws compared to a normal distribution (shown in black). $\lambda = 1$ and $\nu = 0$ correspond to the same white noise distribution. Varying λ only changes the intensity of the pixels, whereas increasing ν increases the spatial correlation length (large structures).

B Supplementary Results

B.1 Spectral Calibration Interventions

We run four experiments in which we manipulate the random fields that are used in the diffusion term of the SDE (*i.e.* the $\boldsymbol{z}$ in the Wiener process $\gamma_\tau \sqrt{\tau} \boldsymbol{z}$, by default it is an i.i.d. Gaussian) and observe the impact on calibration and skill in the spectral domain. First, we rescale the variance of the Wiener process (*i.e.* $\boldsymbol{z} \leftarrow \lambda \boldsymbol{z}$, top row of Figure 11). Then, we vary the noise colour by sampling random fields with a power law PSD (bottom row of Figure 11). Later, we follow (Price et al., 2023) and sample isotropic white noise in the spherical harmonics to respect the geometry of the Earth (Figure 14). Finally, we fit an isotropic parametric noise model to the ensemble error spectra (Figure 17). All experiments occur at inference (no training) and use the same Xaurora checkpoint, producing a 10-member ensemble on 8 initial conditions in 2022 (test set) using the procedure in Appendix A.4.5. We evaluate the ensemble at 12-hour lead time, where the calibration regime is different for mesoscale (underdispersive) and synoptic-scale (overdispersive) variables (see Figures 4 and 31).

Spectral calibration metrics. Using the zonal spectrum of Equation (22), we compute the spectra of the ensemble spread ($P_{\text{spread}}(k)$, k=wavenumber) and of the ensemble mean error ($P_{\text{err}}(k)$), together with the unbiased error spectrum $P^u_{\text{err}}(k) = P_{\text{err}}(k) - P_{\text{spread}}(k)/M$. These tell us which wavenumbers are overdispersed ($P_{\text{spread}} > P_{\text{err}}$) or underdispersed ($P_{\text{spread}} < P_{\text{err}}$) and are related to global spread, uRMSE, and SSR with a cumulative sum:

$$\text{SPREAD} = \sqrt{\sum_k P_{\text{spread}}(k)}, \quad \text{uRMSE} = \sqrt{\sum_k P^u_{\text{err}}(k)}, \quad \text{SSR} = \sqrt{\frac{\sum_k P_{\text{spread}}(k)}{\sum_k P^u_{\text{err}}(k)}}. \tag{29}$$

The numbers differ slightly from the global metrics shown in Section 5 as we use fewer initialisations, but the trends are the same.

B.1.1 No Intervention: White Noise

The white noise ensemble (no intervention) is shown by the grey lines in Figures 12, 13 and 15. The PSD of the ensemble error approximately follows a power law while the PSD of the ensemble spread is flat after a certain wavenumber, which is much lower for mesoscale variables (Q850, U500). That is the noise floor from Section 5.2. Mesoscale variables are underdispersive across all wavenumbers, whereas synoptic-scale variables are overdispersive in the smaller scales, which dominate when integrating the SSR. This is confirmed by the cumulative plots, which show that mesoscale variables are underdispersive across all wavenumbers. Conversely, synoptic-scale variables are underdispersive

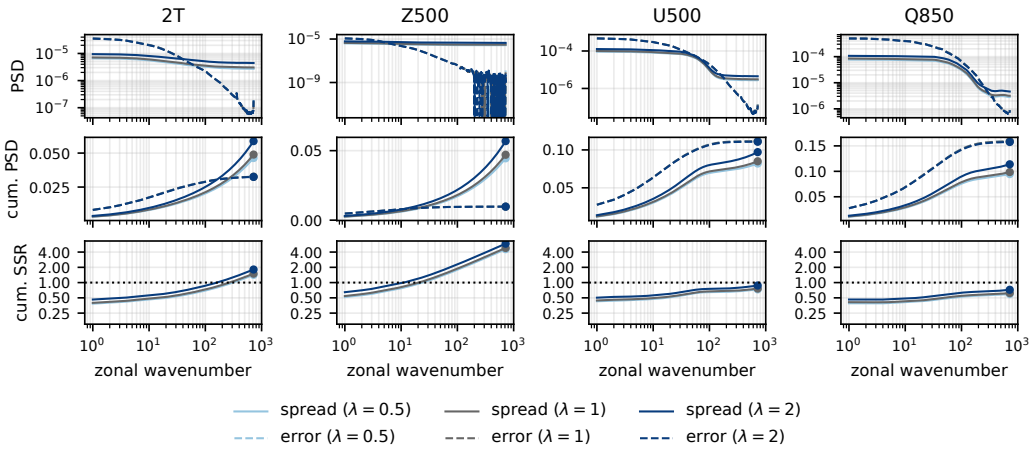

Figure 12: **Spectral calibration of an 8-initialisation** 12 h **ensemble rollout with scaled variance (λ). Top row:** Zonal PSD of the ensemble spread (solid line) and unbiased ensemble mean error (dashed line). Z500 is unstable on large wavenumbers because of the log scale and $P^u_{\text{err}}(k > 200) \approx 0$. **Middle row:** Cumulative PSD of the ensemble spread (solid line) and unbiased ensemble mean error (dashed line). The global spread and uRMSE correspond to the final values (circles). **Bottom row:** Cumulative spectral SSR; the dotted line is perfect calibration and the global SSR corresponds to the final value (circle). Curve colours correspond to different scaling values (λ). White noise corresponds to $\lambda = 1$ (grey). Curves are smoothed with a rolling average of size 7.

when considering only the very large scales (low wavenumbers) and become overdispersive when considering the entire range of scales.

B.1.2 Variance scaling

White noise analysis highlighted that synoptic-scale variables have far too much power in the high-frequency domains, where they are strongly overdispersive. Based on this observation, we experiment with scaling the noise by a scalar λ to assess the effect of less variance ($\lambda = 0.5$) and more variance ($\lambda = 2$), compared to white noise ($\lambda = 1$).

Figure 12 shows that higher variance levels generally mean high spread and lower (or unchanged) error. This indicates that post-hoc calibration is possible to some extent as overdispersive variables (2T, Z500) are both better calibrated and have better error with $\lambda = 0.5$, whereas underdispersed variables (Q850, U500) have better numbers for $\lambda = 2$. However, there are a few caveats to this approach. First, the denoiser is trained on unit-variance random fields, hence changing the variance too much could have unexpected effects. Secondly, a single value of λ cannot fix all variables. The calibration must be tuned per variable. Thirdly and most importantly, the spread increase (or decrease) obtained from scaling the variance is unlikely to be meaningful as it only changes pixel intensity rather than creating dynamically-different members, which is what we expect from ensemble forecasts.

B.1.3 Noise colouring

In an attempt to make the random fields manipulation more meaningful than variance scaling, we experiment with noise colour. To do so, we sample random fields that have a power law-like PSD[12]. This has the effect of creating larger random scale structures, *i.e.* increasing spatial correlation (see Figure 11).

[12]Random fields ($\boldsymbol{\zeta}_\nu$) with $\text{PSD}(k) \propto k^{-\nu}$ are constructed from a standard normal field ($\boldsymbol{\zeta}$) with a radial amplitude filter, $\tilde{\boldsymbol{\zeta}}_\nu = \mathcal{F}^{-1}[A \odot \mathcal{F}[\boldsymbol{\zeta}]]$, with $A(\boldsymbol{k} \neq \mathbf{0}) = \|\boldsymbol{k}\|^{-\nu/2}$ and $\mathcal{F}$ the Fourier transform (2D FFT), followed by unit-variance standardisation, $\boldsymbol{\zeta}_{\nu,ij} = \tilde{\boldsymbol{\zeta}}_{\nu,ij} / \sum_{ij} \tilde{\boldsymbol{\zeta}}^2_{\nu,ij}$

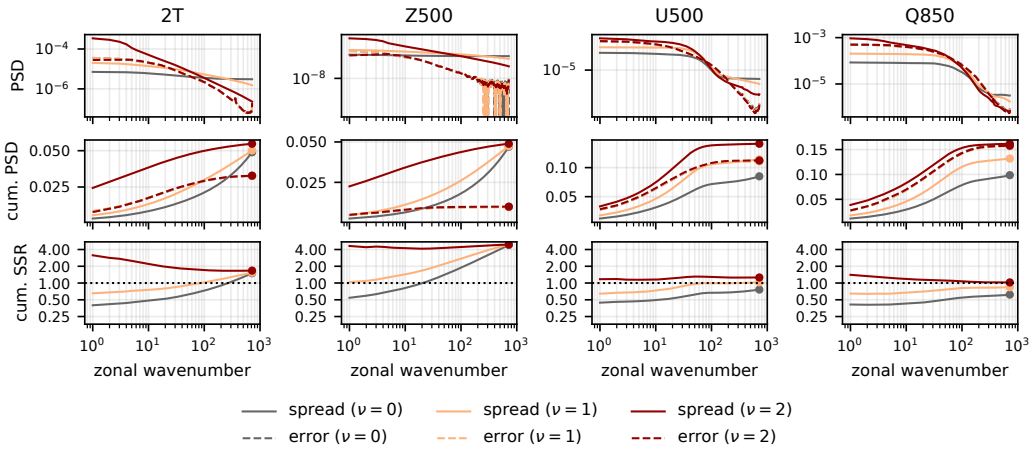

Figure 13: **Spectral calibration of an 8-initialisation** 12 h **ensemble rollout with varying noise colour (ν). Top row:** Zonal PSD of the ensemble spread (solid line) and unbiased ensemble mean error (dashed line). Z500 is unstable on large wavenumbers because of the log scale and $P^u_{\text{err}}(k > 200) \approx 0$. **Middle row:** Cumulative PSD of the ensemble spread (solid line) and unbiased ensemble mean error (dashed line). The global spread and uRMSE correspond to the final values (circles). **Bottom row:** Cumulative spectral SSR; the dotted line is perfect calibration and the global SSR corresponds to the final value (circle). Curve colours correspond to ν exponents. White noise corresponds to $\nu = 0$ (grey). Curves are smoothed with a rolling average of size 7.

Figure 13 shows that changing the colour of the noise has a consistent effect on the PSD of the error and the spread. In terms of tail effects (circles), the spread typically increases, most notably for mesoscale variables, and the error remains roughly constant (marginal increase). Because the starting SSR is different for different variables, we again observe that the optimal ν is variable-dependent; synoptic-scale variables (2T, Z500) cannot be calibrated with the tested values of ν, while U500 and Q850 are almost perfectly calibrated with $\nu = 1$ and $\nu = 2$, respectively. Expanding the analysis to the spectral domain, it appears that increasing ν affects mostly affects the larger scales. It can be seen in the synoptic-scale variables cumulative PSD plots, where the $\nu = 2$ curves lie much more in the top left corner than the white noise curves. The spectral SSR reacts consistently with an increase in lower wavenumbers; with some variables displaying a slope inversion with $\nu = 2$ (2T, Q850). Overall, noise colouring seems more effective than scaling the variance in terms of truly changing the calibration dynamics. Adding spatial correlation in the random fields appears to even out the calibration across wavenumbers, which is an interesting property, since ensembles should be well-calibrated at every scales.

B.1.4 Isotropic spherical white noise

Xaurora uses white noise on a grid that ignores the spherical geometry of the Earth. In this experiment, we follow (Price et al., 2023) and sample white noise on the spherical harmonics to obtain an isotropic white noise that respects geometry (see Figure 14).

The effect of isotropic spherical noise on the global metrics (circles) is a slight increase in SSR explained by a slight increase in spread and a marginally higher error. The PSD plots are particularly interesting because the geometry-aware random fields remove the tail flattening of the spectra (noise floor). Yet, it appears that the miscalibration is not only due to geometrical artifacts, since the global SSR is barely affected.

B.1.5 Noise model fitting with Matérn covariance

Finally, we fit a variable-dependent noise model to address the shortcomings of the other methods. We choose to parameterise the random fields using a Matérn covariance function applied to isotropic Gaussian noise on the sphere using spherical harmonics (similar to (Lindgren et al., 2011)). This

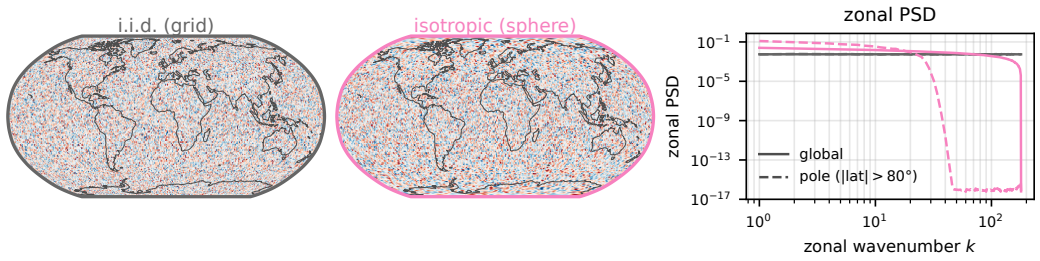

Figure 14: **Noise fields and PSD on a flat grid and on the sphere. Maps:** Fields resulting from sampling white noise on a flat grid (i.i.d. (grid), equivalent to white noise in other experiments, shown in grey) and on the sphere (isotropic (sphere), equivalent to (Price et al., 2023), shown in pink). **Right panel:** Global (solid line) and polar (dashed) PSD of both fields. The grid-based white noise has constant power across wavenumbers and latitudes. Because the earth is a sphere, grid cells in the poles are smaller than on the equator. Therefore, grid-based white noise over-powers the high frequencies in these regions. The spherical version address this problem by explicitly calibrating the variance and spatial correlation to the geometry.

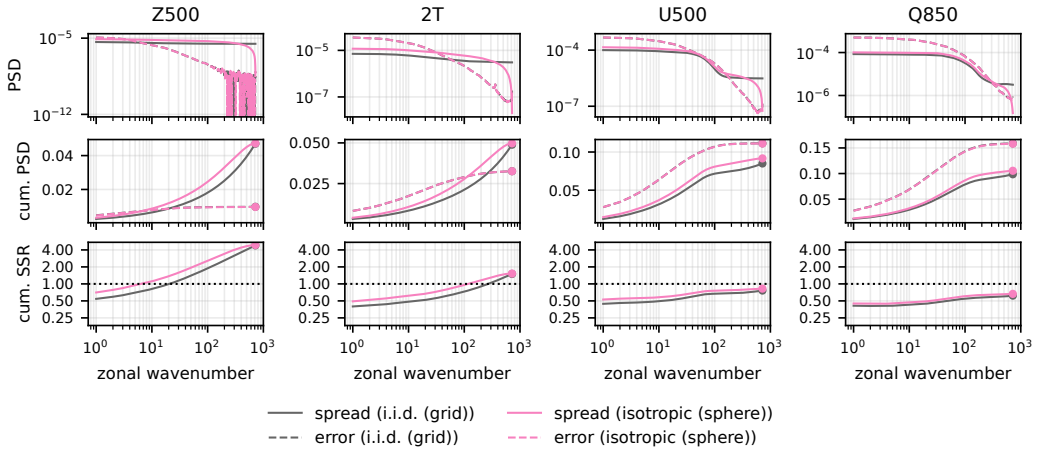

Figure 15: **Spectral calibration of an 8-initialisation** 12 h **ensemble rollout with i.i.d. grid white noise and spherical isotropic white noise. Top row:** Zonal PSD of the ensemble spread (solid line) and unbiased ensemble mean error (dashed line). Z500 is unstable on large wavenumbers because of the log scale and $P^{u}_{\text{err}}(k > 200) \approx 0$. **Middle row:** Cumulative PSD of the ensemble spread (solid line) and unbiased ensemble mean error (dashed line). The global spread and uRMSE correspond to the final values (circles). **Bottom row:** Cumulative spectral SSR; the dotted line is perfect calibration and the global SSR corresponds to the final value (circle). Isotropic spherical white noise is shown in pink, the flat grid version is shown in grey. Curves are smoothed with a rolling average of size 7.

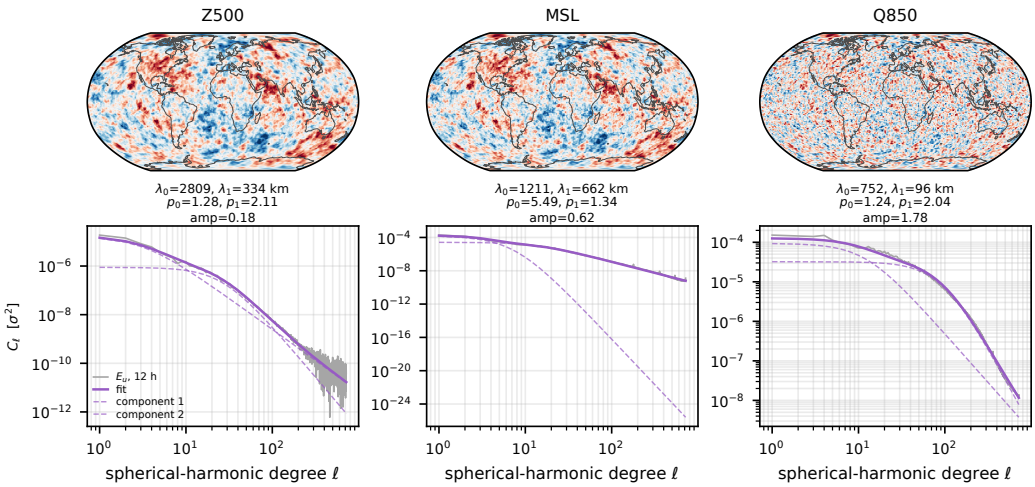

Figure 16: **Random fields fitted to the ensemble error. Top:** Samples drawn from the fitted distribution of different variables. **Bottom:** The PSD of the ensemble mean error (grey) and of the Matérn fit (purple, individual components dashed). Z500 has the most large scale structures (largest λ).

parameterisation is isotropic by construction and has a longer tail than a mixture of SPPT (Palmer et al., 2009) (used in NeuralGCM (Kochkov et al., 2024)).

Fitting and evaluation procedure. We fit a two-component Matérn covariance function, $C_\ell = \sum_{i=0}^{1} F_{0,i}(1 + (\lambda_i/R_e)^2\ell(\ell+1)/2)^{-p_i}$, to the unbiased ensemble-mean error spectrum $P^u_{\text{err}}(\ell)$, generated with white noise (here the spectrum is taken on the spherical harmonics, where ℓ is the degree, rather than the zonal wavenumber k). The correlation length is set by λ_i and the exponents p_i regulate the smoothness (power-law). We fit each variable and 4 pressure levels independently at a single 12 h lead on 8 initial conditions and 10 ensemble members from 2021 (Xaurora validation set). C_ℓ fixes the spectral shape of the noise process, but not the power level. Hence, the amplitude of each channel is calibrated by two fixed-point iterations $\text{amp} \leftarrow \text{amp}/\text{SSR}$, on the same fit period and lead. The fitted spectra are mapped onto the model's 13 levels by nearest neighbour and the constructed random fields are injected into the SDE instead of white noise. The frozen (C_ℓ, amp) are then evaluated on 8 initial conditions from 2022 with a 10-member, 20-step (5-day) rollout (same setup as other spectral experiments, based on Appendix A.4.5). The resulting fields are shown in Figure 16.

Figure 17 shows that fitting the injecting random noise that shares the same spectral properties as the unbiased ensemble mean error successfully calibrates the ensemble across all wavenumbers and variables. However, the gains do not generalise across lead times as the noise fitting is only done at 12 h and no feedback mechanism is implemented (see Figure 18). Fitting the 13 pressure levels separately with a full 3D noise model might help improve the error to some extent, but as stated in the main text, these parameterisations target mostly the large scales that are typically very predictable. Therefore, the model likely relies heavily on them, so training with this sort of random fields would allow the denoiser to properly adapt to these new spectral properties.

B.2 Noise floor investigation

B.2.1 Generative model and solver configuration

A natural hypothesis would be that the noise floor could be caused by a suboptimal choice of generative model or SDE solver. We fix the compute budget to 200 number of function evaluations (NFE) per physical steps and evaluate a first order solver (Euler-Maruyama, EM) against the second order used in Xaurora (Heun). Table 8 details the tested solver configurations. We also compare the DSI and DDPM (see Appendix B.3.1) versions of Xaurora. We use the matched benchmark approach mentioned in Appendix A.4.5. The results show that (1) the **main Xaurora is properly**

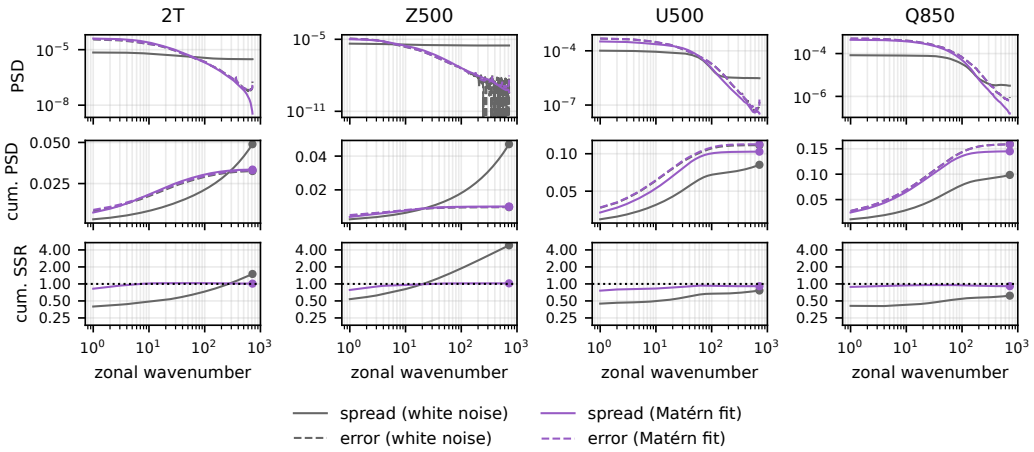

Figure 17: **Spectral calibration of an 8-initialisation** 12 h **ensemble rollout with flat white noise and the fitted Matérn fields. Top row:** Zonal PSD of the ensemble spread (solid line) and unbiased ensemble mean error (dashed line). Z500 is unstable on large wavenumbers because of the log scale and $P^u_{\text{err}}(k > 200) \approx 0$. **Middle row:** Cumulative PSD of the ensemble spread (solid line) and unbiased ensemble mean error (dashed line). The global spread and uRMSE correspond to the final values (circles). **Bottom row:** Cumulative spectral SSR; the dotted line is perfect calibration and the global SSR corresponds to the final value (circle). Matérn fields are shown in purple, the flat grid version is shown in grey. Curves are smoothed with a rolling average of size 7.

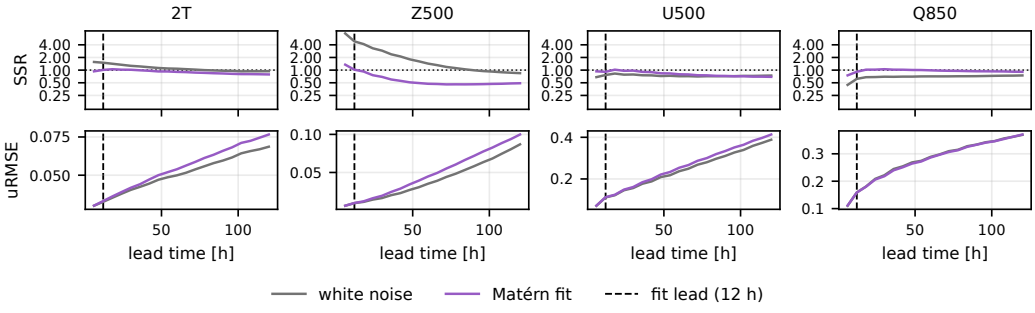

Figure 18: **SSR and uRMSE against lead time for the white noise and fitted Matérn fields. Top:** SSR. **Bottom:** uRMSE. Metrics are shown for different variables, the lead time of the fit is shown by the dashed vertical line (12 h), the white noise curves are in grey and the fitted ones in purple. Generally, the fitted random fields only fix the metrics at the lead time of the fit. The uRMSE is clearly negatively impacted, especially on longer lead times.

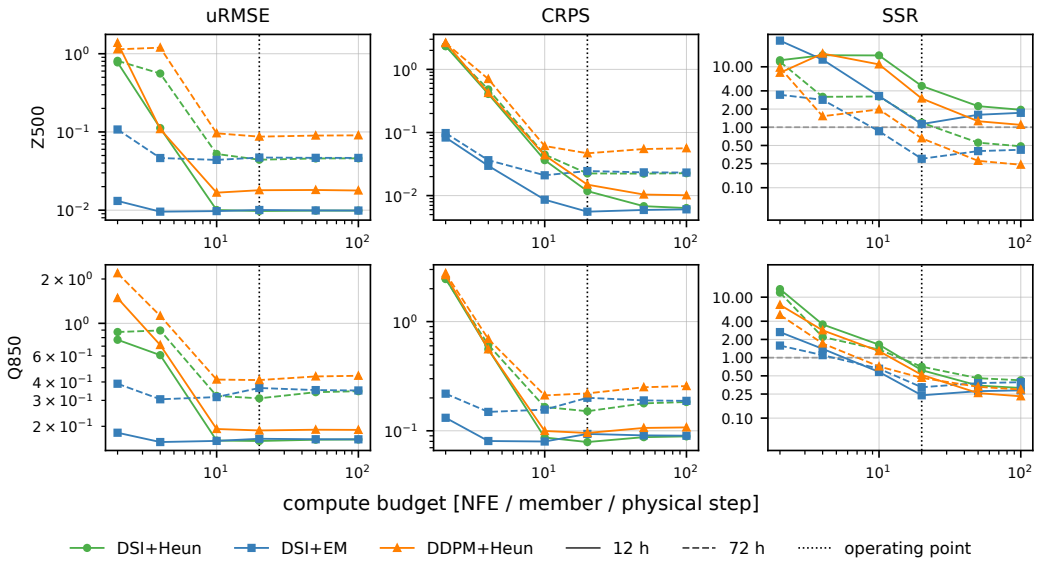

Figure 19: **Convergence for fixed single-member budget.** Unbiased metrics (uRMSE, CRPS, SSR) for Z500 and Q850 at 12 (solid) and 72 (dashed) hours lead time for different generative models and solvers (colours). Xaurora operating 20 NFE per member and physical step is shown in dotted vertical line. The x-axis shows the normalised compute budget (x=10 means 10 integration steps for EM but 5 for Heun).

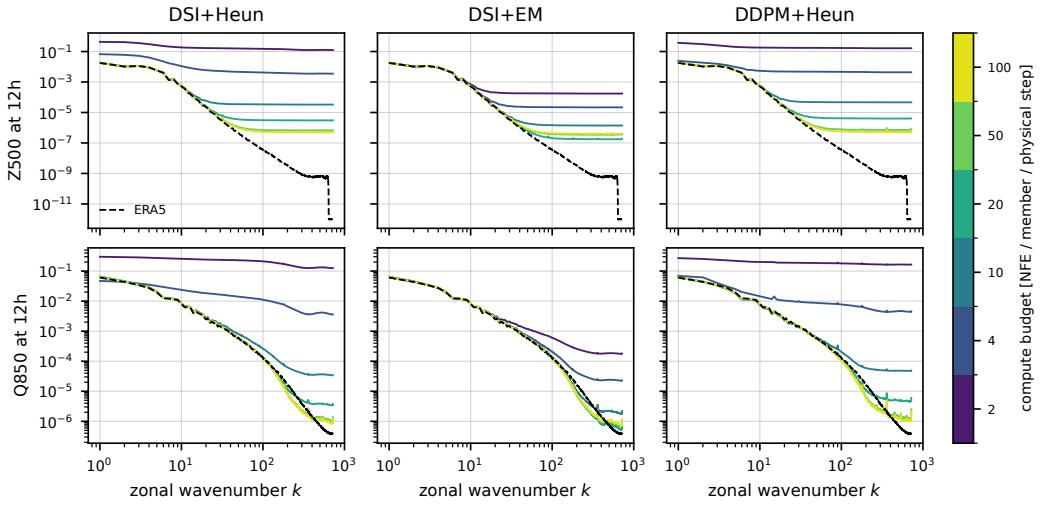

Figure 20: 12 h **PSD averaged across ensemble members for fixed single-member budget.** Ensemble-mean PSD at 12 h for Z500 and Q850 for different generative models and solvers (columns). The operating Xaurora is DSI+Heun with 20 NFE per member and physical steps. The colour bar shows the normalised compute budget (x=10 means 10 integration steps for EM but 5 for Heun).

Table 8: **Hyperparameter grid shared by all three configurations**. The number of SDE steps S follows from the solver's drift evaluations per step (Heun: 2, Euler–Maruyama: 1), hence we use NFE as normalised measure of integration steps. Ensemble sizes (M) are chosen such that every point costs the same, *i.e.* $M \times \mathrm{NFE} = 200$. Only $\mathrm{NFE} = 2$ departs from this, capped at $M = 64$. The main Xaurora configuration used in the paper is $\mathrm{NFE} = 20$, $M = 10$ (Heun, $S = 10$) and is shown in bold.

			SDE steps S	
NFE	M	$M \times$ NFE	Heun	EM
2	64	128	1	2
4	50	200	2	4
10	20	200	5	10
20	**10**	**200**	**10**	20
50	4	200	25	50
100	2	200	50	100

configured and (2) the **noise floor is a property of the denoiser** and not the solver, as it appears for all configurations independently.

Figure 19 shows that the main Xaurora configuration (DSI, 20 NFE, Heun) has converged, *i.e.* it does not get better when increasing the compute budget. Generally speaking, the DDPM parameterisation underperforms in all metrics and lead times, and its performance degrades beyond a compute budget of 10 to 20 NFE, indicating that DSI not only converges faster, but also generates higher quality samples. It could also indicate that the soft constraint to $S = 10$ in the replay buffer algorithm is sufficient for DSI, but not for DDPM. No clear winner can be established between DSI with Heun or Euler-Maruyama, which both achieve their optimal performance around the operating budget of 20 NFEs. The Heun solver was selected early in the development process with the rationale that we could lower the number of integration steps and improve the skill with a second order solver, which is still a reasonable choice.

Figure 20 shows that increasing the NFE reduces the total power of the predictions and improves the fit of the spectra. That is expected, as more SDE steps generally give more room for denoising. In addition, the noise floor appears in all configurations. Because the DSI and DDPM are entirely different runs, their denoisers are independent, indicating that the noise floor is a property of the denoiser rather than the SDE solver or the training algorithm. Note that with the Euler-Maruyama solver, the DSI noise floor starts worsening (higher power and lower cut-off wavenumber) after 20 NFE.

B.2.2 Sufficient denoiser training

Another natural hypothesis would be that the denoiser might be under-trained and therefore fails to remove noise on all wavenumbers. We evaluate different Xaurora checkpoints and record the average PSD of the predictions (8 initialisations, 2 ensemble members using the procedure in Appendix A.4.5). The results in Figure 21 refute the under-training hypothesis. The noise floor remains essentially constant (slight increase for Q850) throughout the training. This suggests that the noise floor is an intrinsic property of the denoiser, rather than a learnt (or *not forgotten*) trait. Whether this comes from the Aurora prior itself cannot be settled by the current analysis.

B.2.3 Contribution of the pretrained prior

To evaluate the contribution of the pretrained prior, we plot the PSD obtained with the denoiser trained from scratch. This model is severely undertrained, yet Figure 22 shows that it also yields a noise floor, although with a slightly lower noise level (power). Furthermore, Figure 23 shows very similar results when using the large version of Aurora as prior.

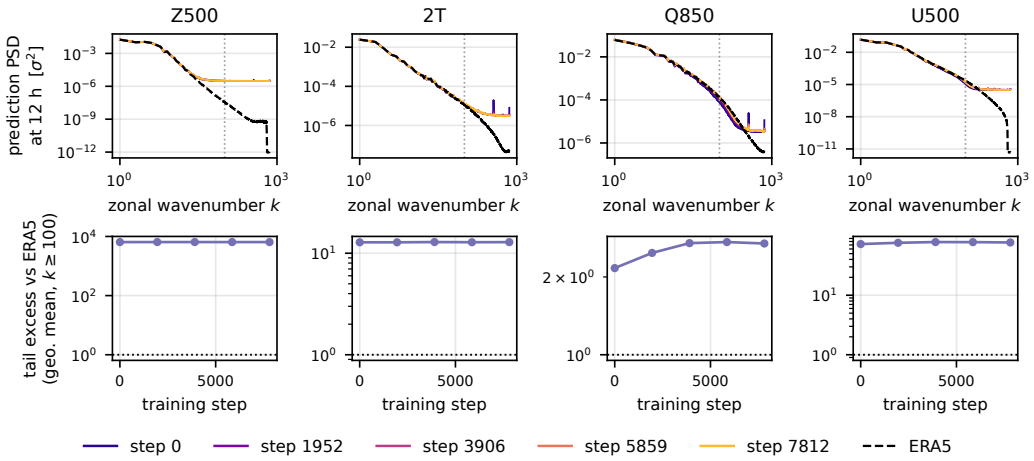

Figure 21: **Spectral properties of different Xaurora training checkpoints. Top:** 12 h PSD averaged across 8 test set initialisations and 2 ensemble members, colours correspond to different checkpoints from initialisation (step 0) to the end of the training (step 7812). Matching ERA5 average spectra are shown with a dashed line. **Bottom:** Average power for wavenumbers greater than 100. This measures the flattening of the tail (*i.e.* the noise floor) of the PSD for each checkpoint.

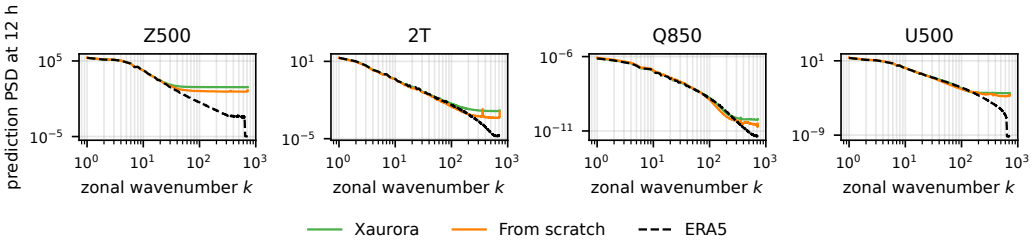

Figure 22: **PSD of Xaurora and denoiser trained from scratch.** 12 h PSD averaged across 8 test set initialisations and 10 ensemble members. Xaurora is shown in green and the denoiser trained from scratch in orange. Matching ERA5 average spectra are shown with a dashed line.

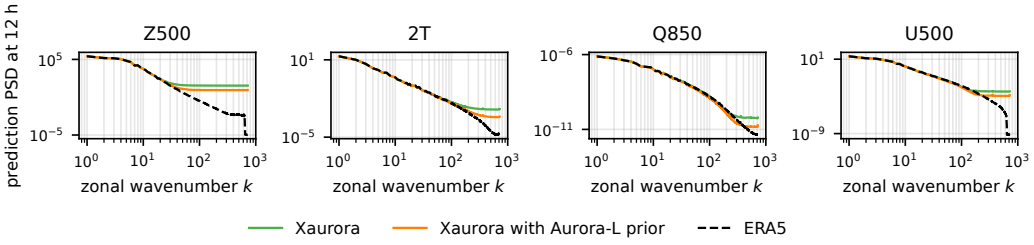

Figure 23: **PSD of Xaurora initialised with different priors.** 12 h PSD averaged across 8 test set initialisations and 10 ensemble members. Xaurora (official, prior is Aurora-S) is shown in green and Xaurora with Aurora-L as the prior is in orange. Matching ERA5 average spectra are shown with a dashed line.

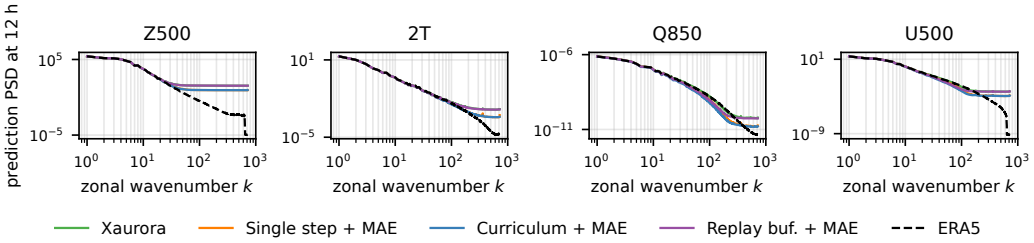

Figure 24: **PSD of Xaurora and other optimisation setups.** 12 h PSD averaged across 8 test set initialisations and 10 ensemble members. Xaurora is shown in green and other colours denote the different optimisation approaches discussed in the ablations. Matching ERA5 average spectra are shown with a dashed line.

B.2.4 Contribution of the finetuning method

We evaluate the effect of our probabilistic rollout finetuning approach on the noise floor by plotting the PSD of Xaurora together with those of the optimisation methods used in the ablations (see Appendix B.3.1 for more details). It appears that all methods produce a clear noise floor, although the level varies. Interestingly, the mid-range wavenumbers are under-powered by the MAE-based methods for the synoptic-scale variables, whereas Xaurora nicely reconstructs ERA5 on those frequencies (see Figure 24).

B.3 Ablations

B.3.1 Main paper ablations details

All ablation experiments in Section 5.3 run with the same hyperparameters as the best performing Xaurora model (see Table 6), except for the changes detailed below.

Optimisation methods. The optimisation of Xaurora is based on an SDE-rollout replay buffer and a CRPS loss. We compare this setup with different configurations:

- **Single-step + MAE:** No rollout training. We apply Algorithm 1 with the Aurora MAE loss (variable weighting, no latitude-weighting) (Bodnar et al., 2025). This model quickly overfits, achieving its best checkpoint after only 200 backward steps (12,800 samples). Nevertheless, the numbers in Table 1 suggest that this setup is relatively close to Xaurora. This is shown to be false in Appendix B.3.2.
- **Curriculum + MAE:** Curriculum rollout training approach similar to (Lang et al., 2024b). Starting from the best checkpoint of the "Single-step + MAE" experiment, we perform a two-stage finetuning: (1) 2-step (12 h forecast) rollout training for 125,000 samples, and (2) linearly increasing rollout steps from 2 to 12 (12 h to 72 h) over 125,000 samples. To make N-step rollout training computationally feasible, we performed $N-1$ gradient-less steps by estimating samples using the denoiser $\boldsymbol{D}_\theta(\boldsymbol{X}_0^t, \boldsymbol{h}^t, \tau{=}0) \approx \mathbb{E}[\boldsymbol{x}^{t+\Delta t} \mid \boldsymbol{h}^t]$, instead of solving the SDE. We then supervise the last step according to Algorithm 1. This mechanism was efficient, but resulted in poor probabilistic performance as the conditional expectation approximation did not help the model generalise to the true inference distribution. This motivated the introduction of the replay buffer.
- **Replay buf. + MAE**: the replay buffer combined with an MAE loss. In this case, we store a single trajectory in the replay buffer entries that we supervise with the Aurora MAE loss. The results were comparable to the "Curriculum + MAE" experiment, highlighting the importance of the probabilistic loss.

Parameterisations. Xaurora leverages the Denoising Stochastic Interpolants introduced in this paper. As discussed in Section 4.2, other parameterisations are unlikely to perform well in the context of finetuning from a pretrained deterministic prior. We also expect the replay buffer mechanism to worsen the performance of the other parameterisations as early predictions do not fully benefit from the Aurora warm-start. Nevertheless, we train and evaluate the following Xaurora variants with the same optimisation (Replay buffer and CRPS):

- $\boldsymbol{x}_1$**-pred. + DDPM**: The diffusion equivalent of DSI. This parameterisation also targets the terminal state distribution and can therefore be properly initialised from Aurora, hence it shares most of the advantages of DSI in the context of finetuning deterministic priors. Yet, it falls short of Xaurora. We hypothesise that the pretrained encoder heads used to embed the SDE state could be more difficult to train with diffusion, which feeds full noise states, than SI, which could lead to a worse or less stable local minimum. Additionally, the replay buffer algorithm depends on the number of integration steps, which is set to $S = 10$, relatively low for diffusion models, potentially making the training of the DDPM network even more challenging.
- $\boldsymbol{W}$**-pred.**: The parameterisation detailed in Appendix A.5 with $\boldsymbol{W}_\theta$ initialised to Aurora-S. The model is optimised with Equation (25). The training is extremely unstable, reaching infinite gradients after roughly 70 optimisation steps. Therefore, we do not run the evaluation, hence the empty row in Table 1. Note that the instabilities could potentially be fixed by adding stabilisation layer norms in the encoder[13] or by training the decoder from scratch, but this would prevent warm-starts ($\boldsymbol{D}_{\theta^{\text{init}}} \neq \boldsymbol{G}_\phi$), confirming that DSI is the superior parameterisation.
- $\boldsymbol{B}$**-pred.**: The vanilla SI parameterisation where the neural network predicts the drift term of the SDE, with $\boldsymbol{B}_\theta$ initialised to Aurora-S. The network is trained on Equation (24). This parameterisation is stable, but does not perform well, especially on probabilistic metrics. At the first validation, Xaurora has an average CRPS of 0.1 while this parameterisation only achieves 1.8, highlighting the superiority of the DSI for finetuning from deterministic priors.

Aurora-L prior. A central result of Xaurora is that it is able to outperform its prior. We show that this property scales to an Aurora-L initialisation. We initialise a Xaurora model from the 1.3B parameter Aurora version and train it with "Single-step + MAE". The resulting model outperforms its prior, Aurora-L, as shown in Table 1. It is worth noting that the number of training steps is kept constant and this large version of Xaurora has not converged. Based on these results and the optimisation ablation, we expect a probabilistic replay buffer version of the large model to perform even better.

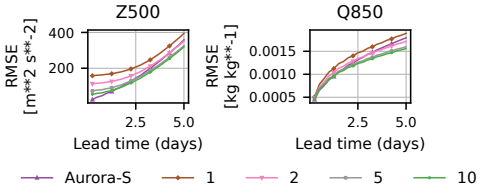

Figure 25: **Effect of the ensemble size on the biased ensemble mean RMSE.** 5-day rollout evaluation of the RMSE (no ensemble size correction) for Z500 and Q850 and ensemble sizes of 1, 2, 5 and 10. With $M = 1$, the 6 h RMSE of Z500 is higher than the full ensemble. Q850 behaves as expected, the RMSE is consistently better with larger M. At a lead time of 5 day, $M = 2$ surpasses Aurora-S.

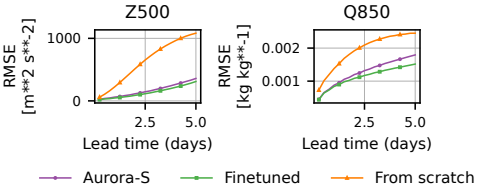

Figure 26: **Effect of training from scratch vs finetuning on the ensemble mean uRMSE (unbiased).** 5-day rollout evaluation of the uRMSE for Z500 and Q850 after training from scratch or finetuning from Aurora-S weights. The finetuned model performs much better. Both models were trained with "Single-step + MAE" (see Appendix B.3.1) because rollout training requires meaningful predictions at initialisation.

B.3.2 In-depth comparison: Xaurora vs Single step + MAE

Figure 27 clearly shows that Xaurora is better than Single step + MAE, especially on longer lead times (from 6 to 7 days the CI bands are mostly below 0, indicating that Xaurora is significantly better). The short term calibration is similar, but the SSR stays significantly closer to 1 with increasing lead times.

B.3.3 Ensemble size

Throughout this paper, we compare Xaurora ensemble mean to deterministic baselines, like Aurora-S or Aurora-L. This is technically not a fair comparison, as an ensemble has a statistical advantage

[13] As suggested in the official Aurora finetuning documentation: https://microsoft.github.io/aurora/finetuning.html#exploding-gradients.

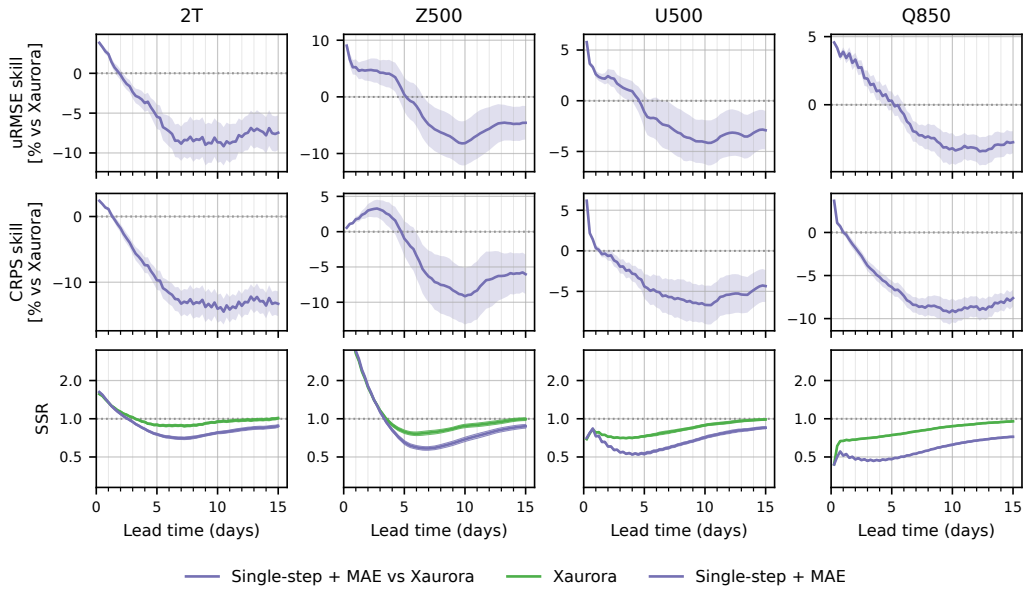

Figure 27: **15-day ablation Xaurora vs Single step + MAE. Top:** uRMSE skill score of Single step + MAE compared to Xaurora; positive values mean that Single step + MAE is better than Xaurora, negative means Xaurora is better. **Middle:** Same for CRPS skill score. **Bottom:** SSR of Single step + MAE and Xaurora. Columns are different variables. Bands are 95% confidence intervals (see Appendix A.4.2).

over a deterministic prediction, which is one of the reason why ensemble models are favoured over deterministic models. For completeness, we show the effect of the ensemble size on the error metrics. As expected, Figure 25 shows that the RMSE (without ensemble size correction, otherwise the effect we try to meaasure is removed) of Q850 (mesoscale representative) and Z500 (synoptic-scale representative) is better with more ensemble members. The poor short lead time performance in Z500 is discussed in the main paper (see Section 5.1). Nevertheless, with 2 members, Xaurora is already better than Aurora-S at 5 days.

B.3.4 Finetuning vs training from scratch

One benefit of Xaurora is the deterministic prior provided by Aurora-S. Figure 26 shows a clear advantage for the finetuned model across lead times. These results should be treated carefully as a fairer comparison would add the number of samples seen during the pretraining of Aurora-S to the training from scratch, but we cannot afford this for an ablation.

B.3.5 Naive probabilistic Aurora baseline: MC-DropPath

As Aurora is trained with stochastic depth (DropPath), we build a naive probabilistic baseline by keeping the mechanism on during inference, i.e., *MC-DropPath* inspired by (Gal & Ghahramani, 2016). Figure 28 validates that Xaurora is more valuable than this simple baseline.

B.4 Additional metrics

Figures 29, 30, 31, 32 and 33 display the RMSE (uRMSE for ensemble models), CRPS, SSR, and PSD (2-day and 10-day) for all the recorded variables. Figures 34 and 35 show calibration maps for a selection of variables at 12 h and 5 days. Finally, Figure 36 and Figure 37 show spatially pooled metrics and rank histograms, respectively.

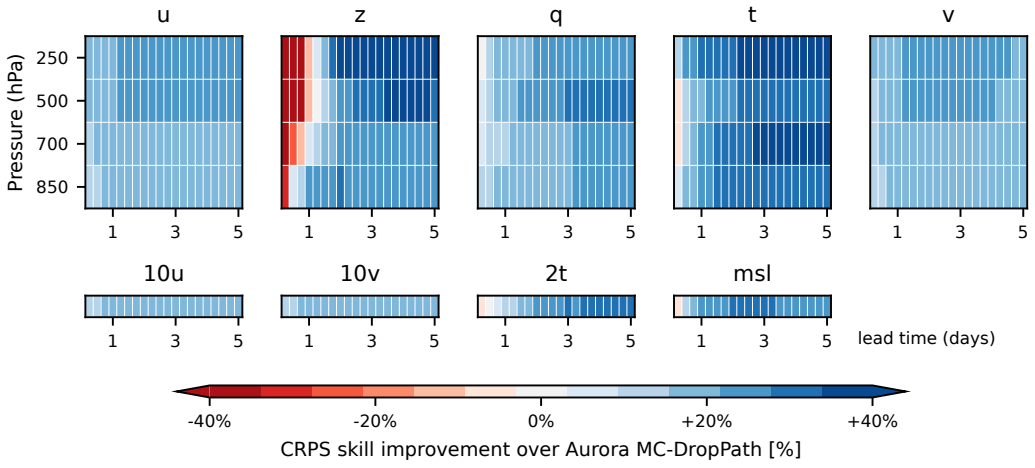

Figure 28: **Xaurora vs Aurora MC-DropPath CRPS scorecard.** Each panel shows the CRPS improvement percentage from Xaurora over a naive probabilistic Aurora-S for one variable across lead times and pressure levels (for atmospheric variables). Both models have 10 ensemble members. Red indicates degraded performance (negative improvement), blue shows improvement over the reference.

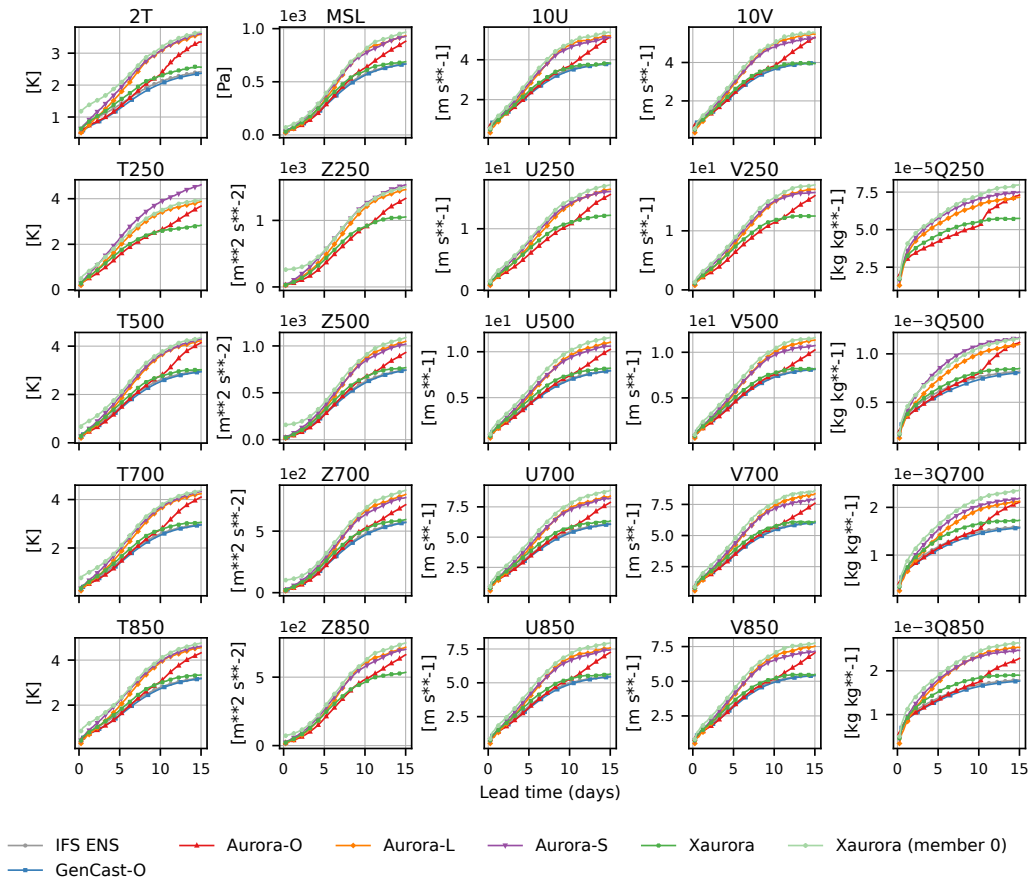

Figure 29: **Global medium-range weather forecasting RMSE (uRMSE for ensemble models) for all recorded variables in 2022 at** $0.25°$**.**

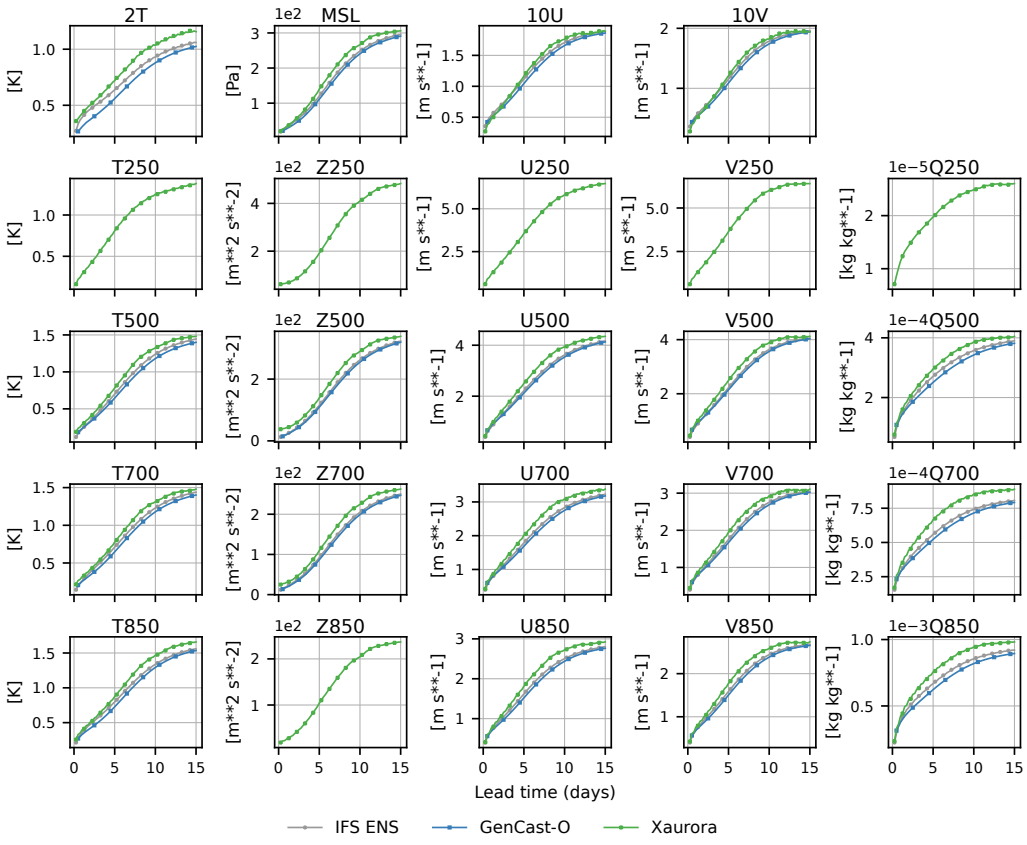

Figure 30: **Global medium-range weather forecasting CRPS for all recorded variables in 2022 at** $0.25°$**.**

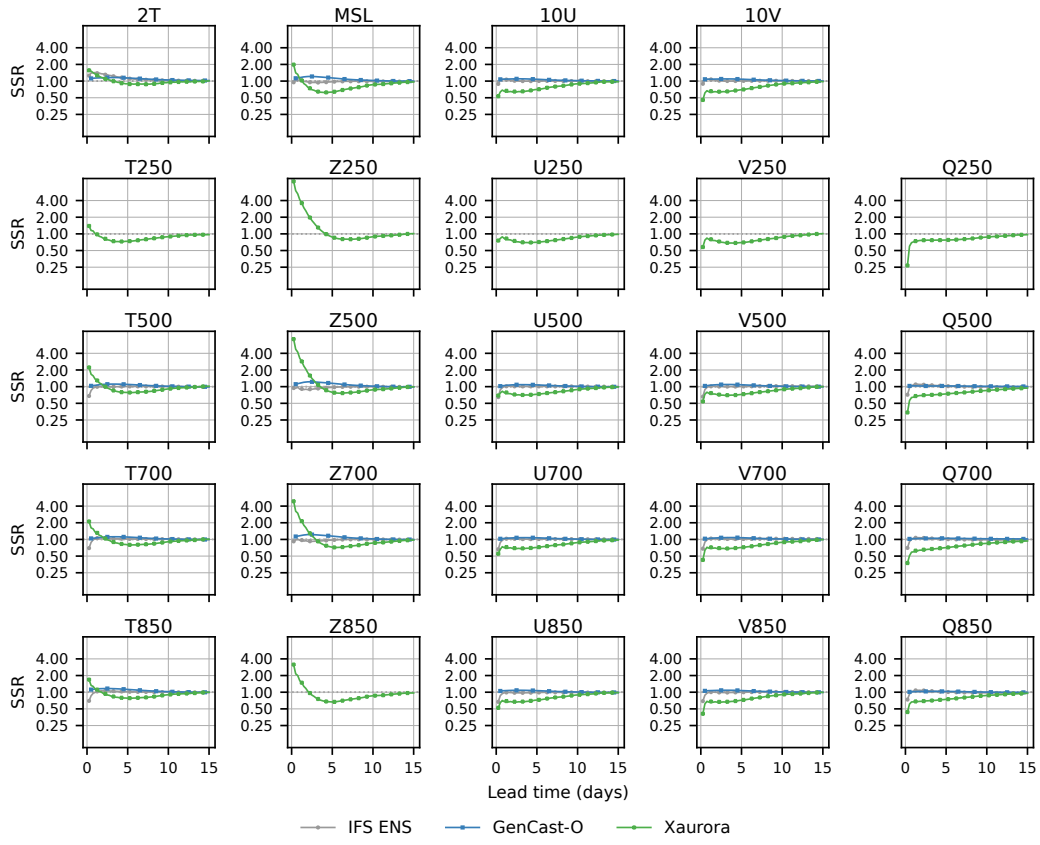

Figure 31: **Global medium-range weather forecasting SSR for all recorded variables in 2022 at** $0.25°$**.**

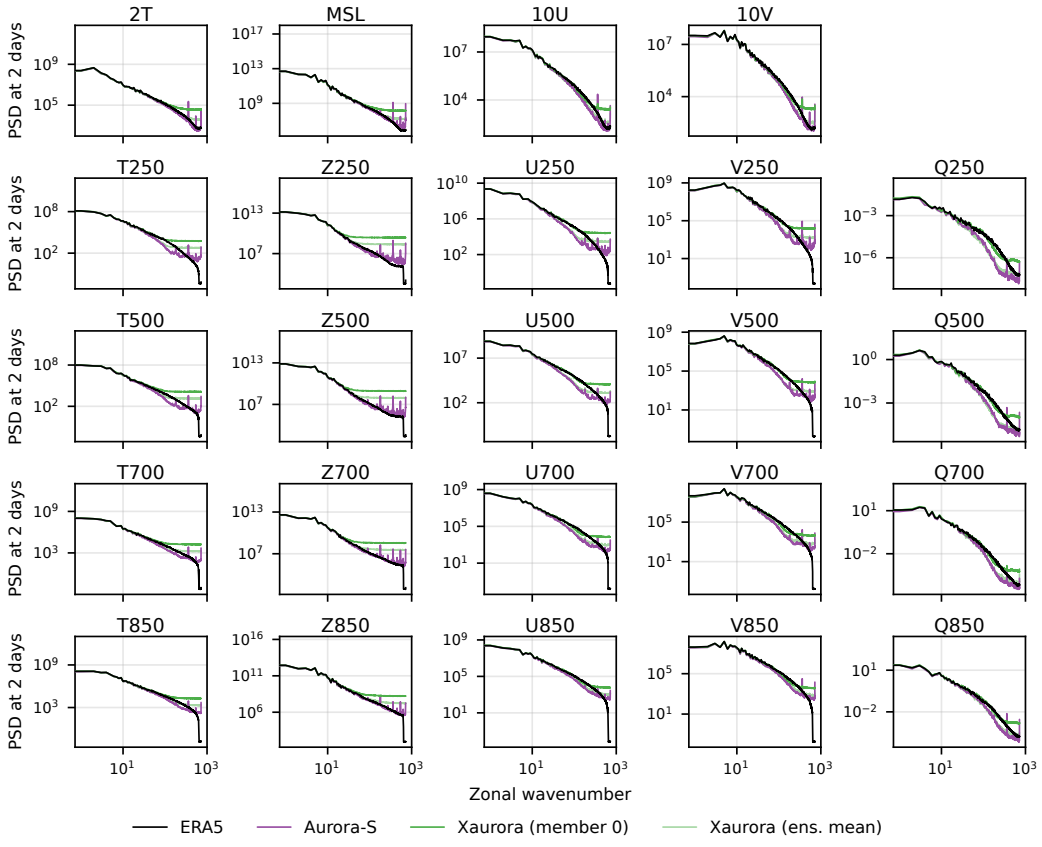

Figure 32: **2-day PSD for all recorded variables in 2022 at** $0.25°$**.**

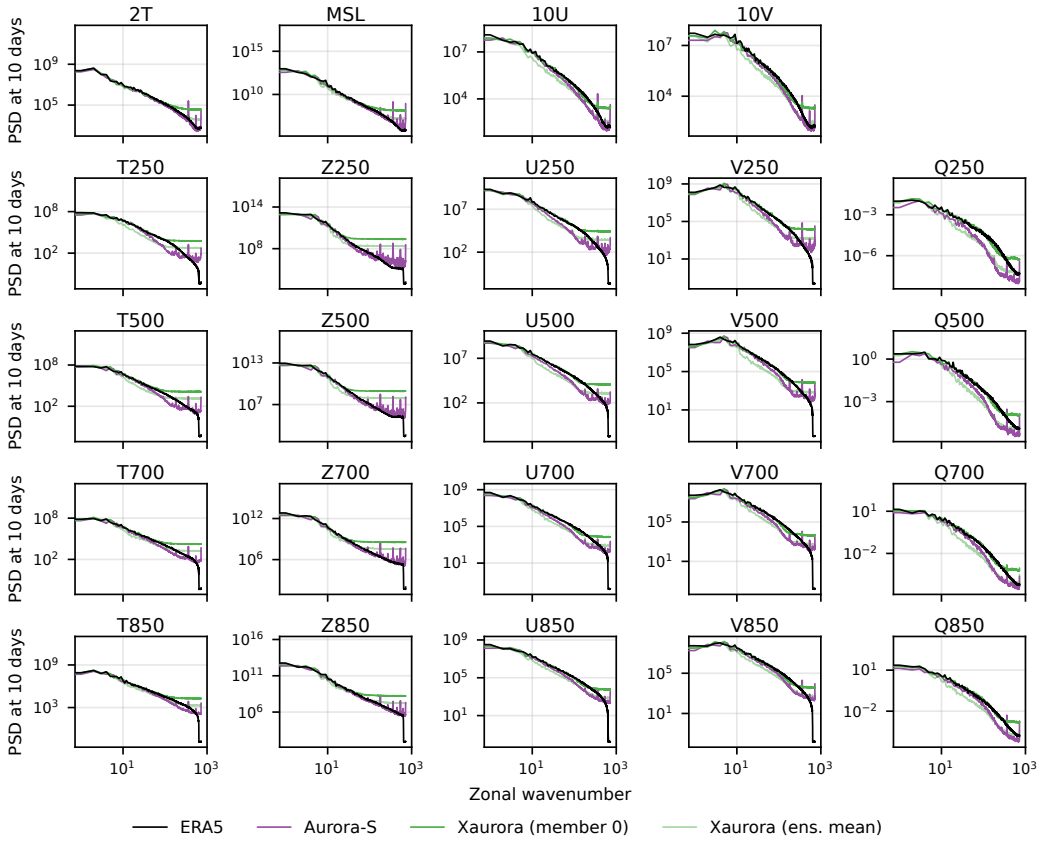

Figure 33: **10-day PSD for all recorded variables in 2022 at** $0.25°$**.**

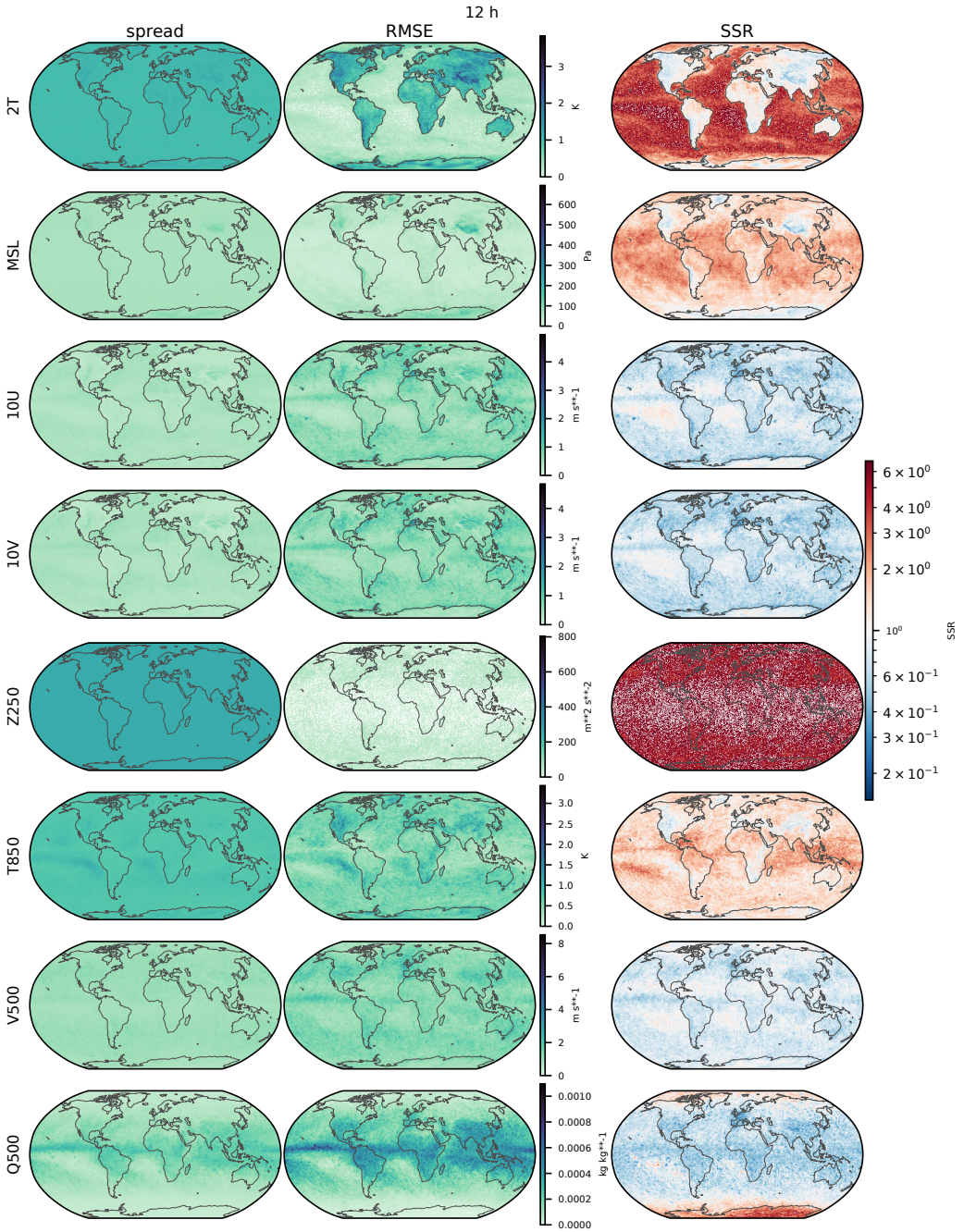

Figure 34: **Calibration maps at** 12 h **for a selection of variables.**

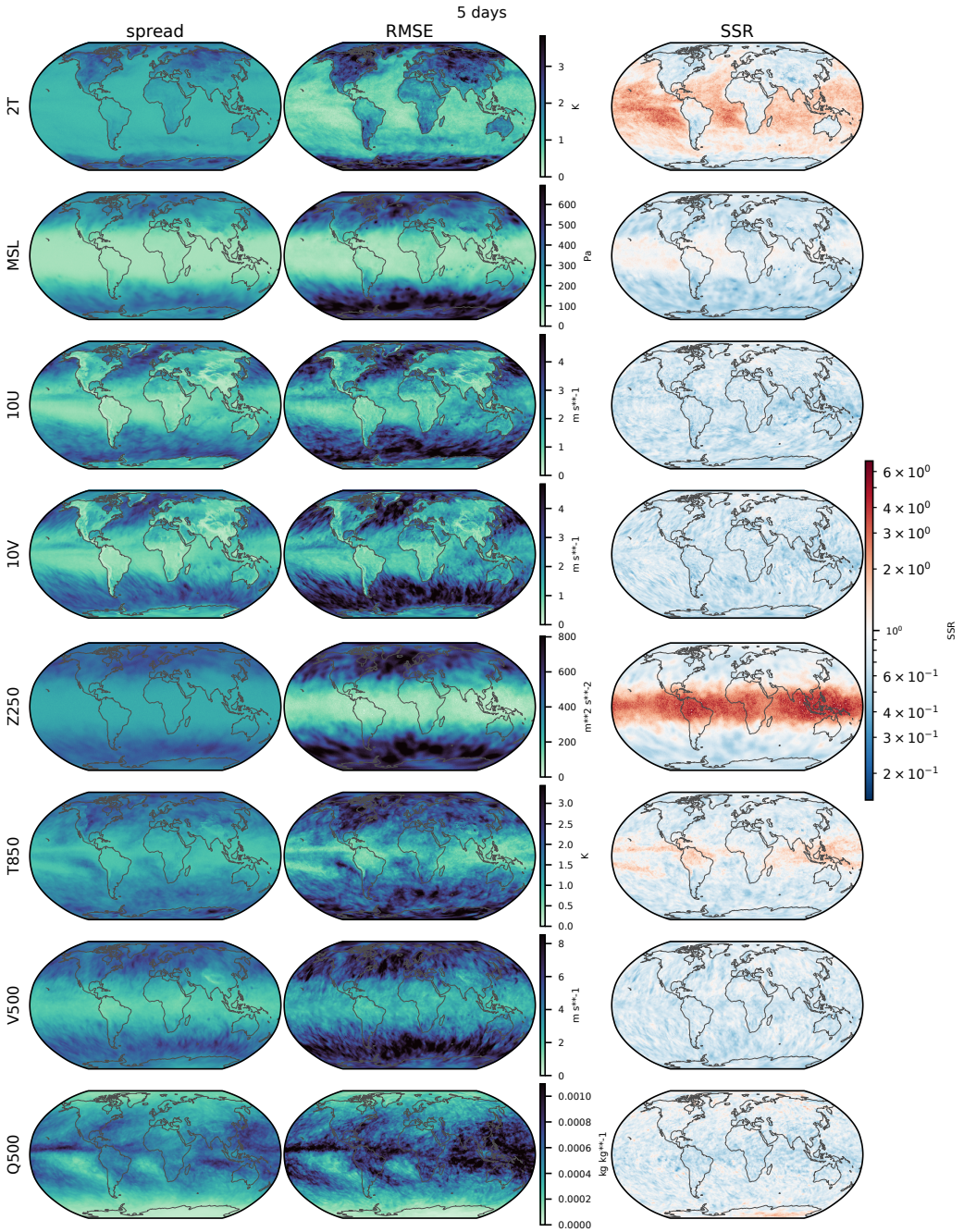

Figure 35: **Calibration maps at 5 days for a selection of variables.**

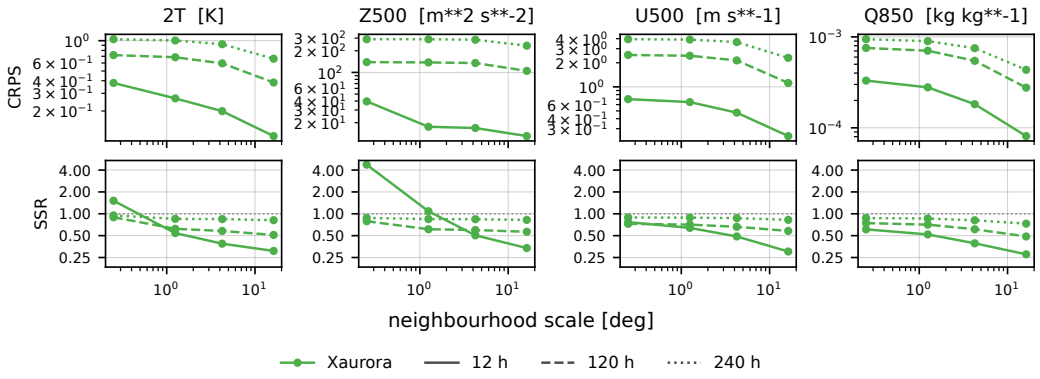

Figure 36: **Spatially-pooled calibration metrics at different lead times.** Across lead times (shown in different line styles), pooling generally improves the CRPS score (top row) and tend to decrease the SSR (bottom row).

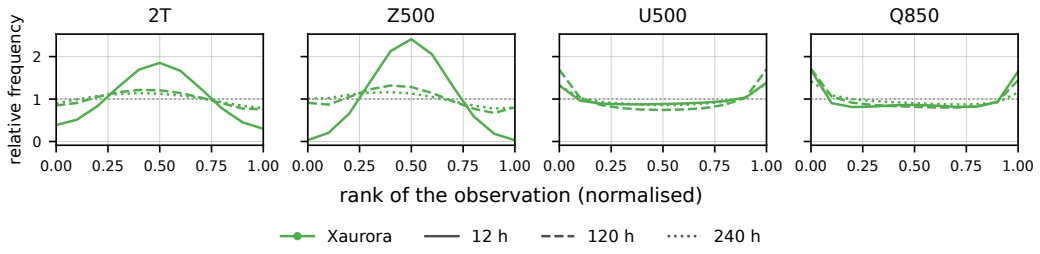

Figure 37: **Rank histograms at different lead times.** 2T and Z500 are overdispersive on short lead times (concave) as highlighted by the rest of the analysis. U500 and Q850 are underdispersive (convex).

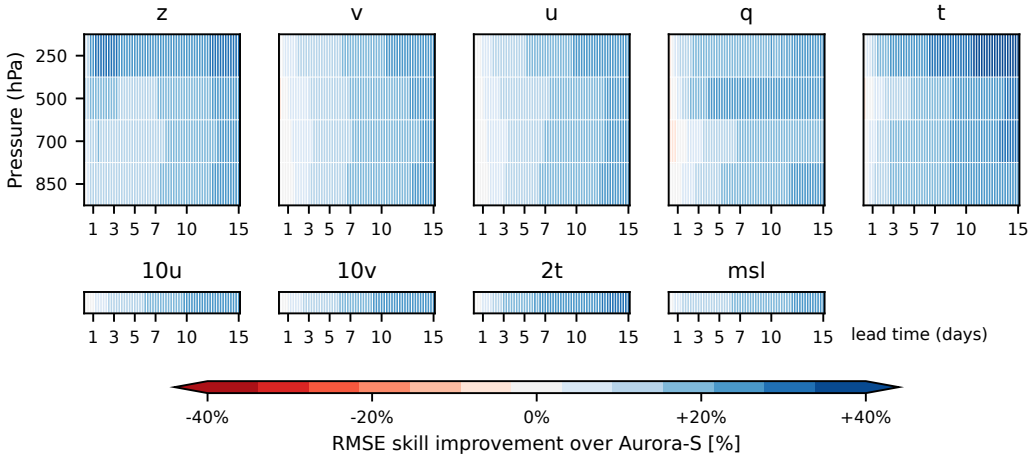

Figure 38: **Xaurora vs Aurora-S RMSE scorecard.** Each panel shows the RMSE improvement percentage from Xaurora over Aurora-S for one variable across lead times and pressure levels (for atmospheric variables). We use uRMSE for Xaurora (unbiased) and the plain RMSE for Aurora-S. Red indicates degraded performance (negative improvement), blue shows improvement over the reference.

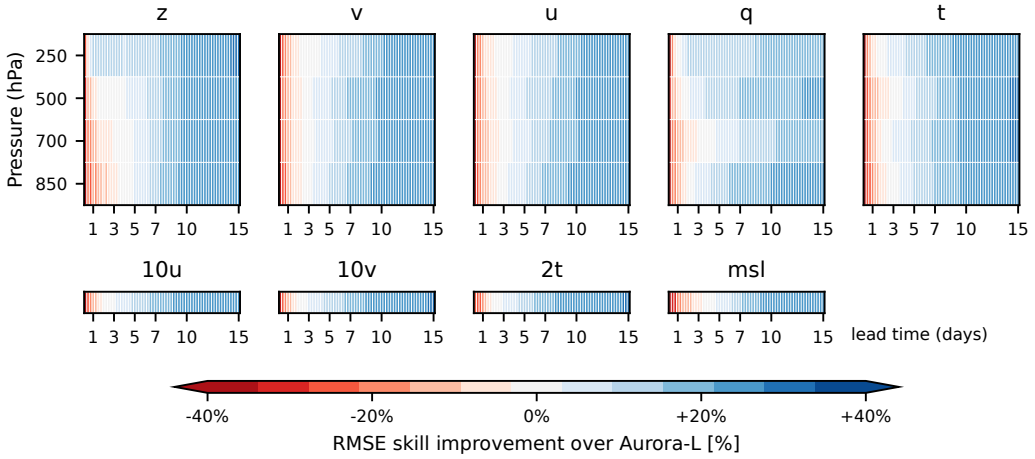

Figure 39: **Xaurora vs Aurora-L RMSE scorecard.** Reads as Figure 38

B.5 Additional comparison with Aurora models

B.5.1 Aurora scorecards

Scorecards comparing Xaurora with Aurora-S (Figure 38) and Aurora-L (Figure 39) show the superiority of our model for medium-range weather forecasting. The benefits of Xaurora over Aurora-S are mostly apparent after 1 day, where most variables and lead times are outperformed. Aurora-L is generally better on slightly longer lead times (around 3 days), after which it falls short against Xaurora. We voluntarily use the uRMSE (approximating the infinite-member ensemble mean error) since we want to characterise the maximal gain from Xaurora over Aurora models. An ensemble-size dependent analysis is provided in Section 5.1 and Appendix B.3.3.

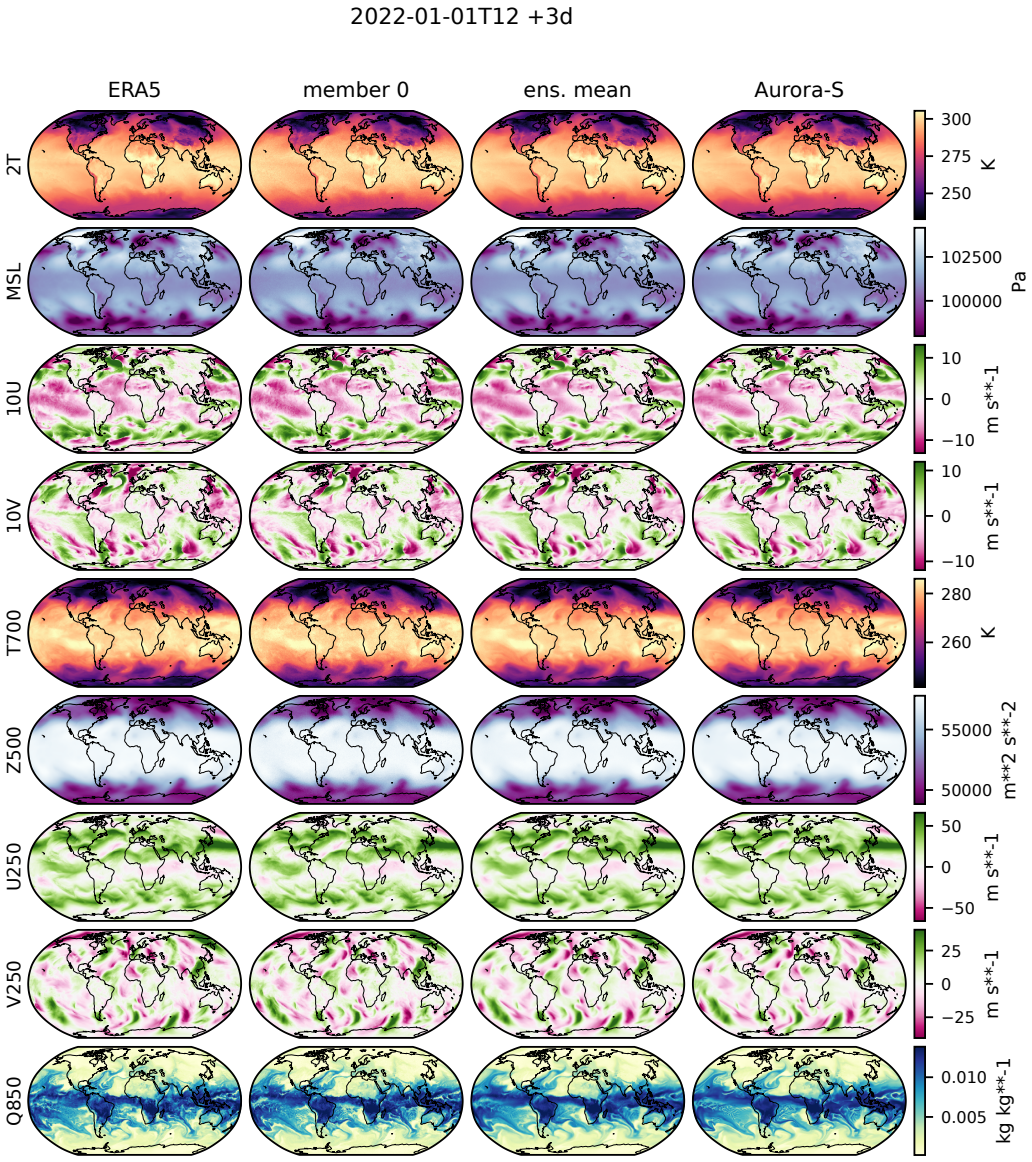

Figure 40: **3-day rollout initialised on January 1st, 2022 at 12:00 UTC.** Columns correspond to ERA5, Xaurora first member (member 0), Xaurora ensemble mean (10 members), and Aurora-S. Variables (unnormalised) are shown on each row.

B.5.2 Forecast visualisation

Figures 40 and 41 show a wide selection of variables and pressure levels for a forecast initialised on 01/01/2022 at 12:00 UTC.

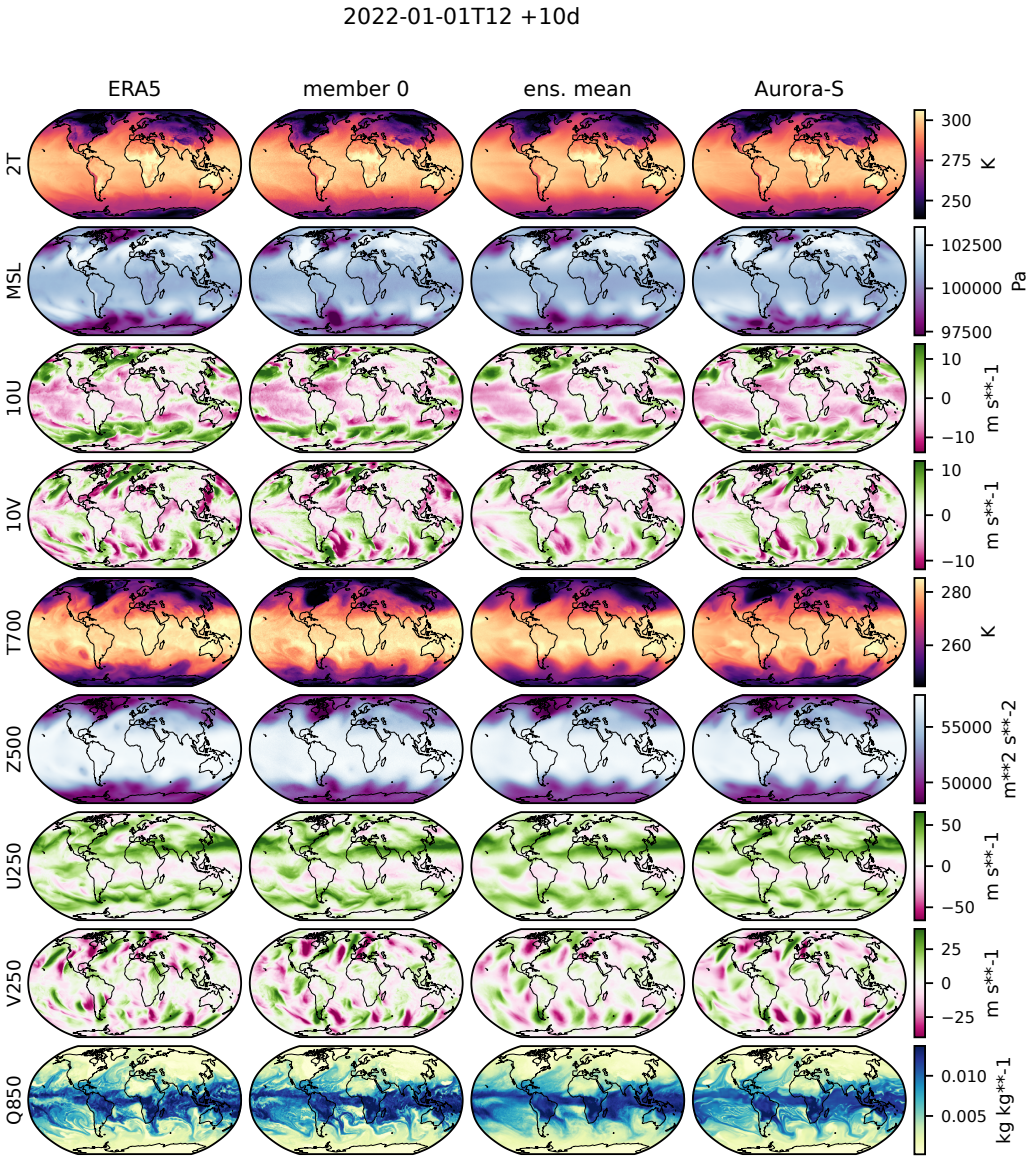

Figure 41: **10-day rollout initialised on January 1st, 2022 at 12:00 UTC.** Columns correspond to ERA5, Xaurora first member (member 0), Xaurora ensemble mean (10 members), and Aurora-S. Variables (unnormalised) are shown on each row.

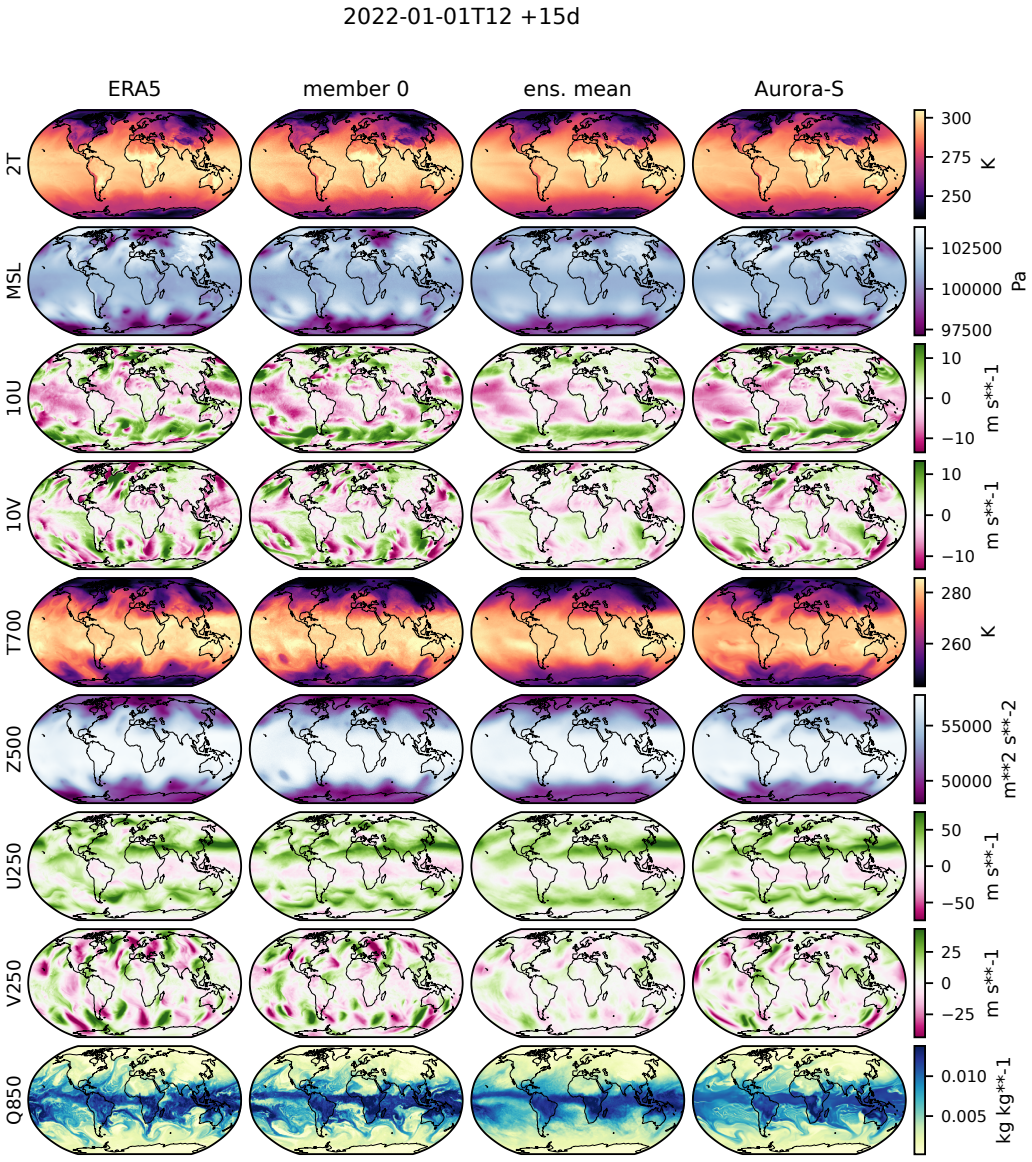

Figure 42: **15-day rollout initialised on January 1st, 2022 at 12:00 UTC.** Columns correspond to ERA5, Xaurora first member (member 0), Xaurora ensemble mean (10 members), and Aurora-S. Variables (unnormalised) are shown on each row.

Algorithm 3 Replay-buffer SDE-rollout training with 2-member fair CRPS

Input: $(\alpha_\tau, \beta_\tau, \gamma_\tau)$, Training data $\mathcal{D}_{\text{train}}$, pretrained denoiser $\boldsymbol{D}_\theta$, training samples N_{samples}, optimiser, buffer capacity N_{B}, injection probability p_{inject}, SDE steps S, depth cap $K_{\max}$, step jitter j, SDE Solver, Replay buffer $\mathcal{B}$.

Simplifying notations: entry $e = (\boldsymbol{X}^{(a)}, \boldsymbol{X}^{(b)}, \boldsymbol{h}^{(a)}, \boldsymbol{h}^{(b)}, \tau, k, t)$ holds the SDE state and histories of two members (a, b) at pseudo-time τ, rollout depth k, and physical time t. $\mathrm{sg}(\cdot)$ is the stop-gradient operator, $\bar{m}$ is the member other than m. When $\tau > 0$, SDE states and history are model-generated.

$\mathcal{B} \leftarrow \varnothing$
for $n = 1$ **to** N_{samples} **do**
  // Sample entry: from the dataset or the buffer
  if $|\mathcal{B}| < N_{\text{B}}$ **or** $U[0,1] < p_{\text{inject}}$ **then**
    $(\boldsymbol{x}^{t-\Delta t}, \boldsymbol{x}^{t}, \boldsymbol{x}^{t+\Delta t}) \sim \mathcal{D}_{\text{train}}$
    if $|\mathcal{B}| = N_{\text{B}}$ **then**
      evict oldest entry from $\mathcal{B}$
    end if
    $e \leftarrow (\boldsymbol{X}^{(a)}{=}\boldsymbol{X}^{(b)}{=}\boldsymbol{X}^{t},\ \boldsymbol{h}^{(a)}{=}\boldsymbol{h}^{(b)}{=}\boldsymbol{h}^{t},\ \tau{=}0,\ k{=}0,\ t)$
  else
    $e \leftarrow$ pop a uniformly random entry from $\mathcal{B}$
    Fetch target $\boldsymbol{x}^{t}$ from dataset $\mathcal{D}$
  end if

  // Supervision: the only gradient-tracked forward
  $m \sim U\{a, b\}$ (select supervised member)
  $\boldsymbol{D}^{(m)} \leftarrow \boldsymbol{D}_\theta(\boldsymbol{X}^{(m)}, \boldsymbol{h}^{(m)}, \tau)$ (gradient)
  $\boldsymbol{D}^{(\bar{m})} \leftarrow \mathrm{sg}(\boldsymbol{D}_\theta(\boldsymbol{X}^{(\bar{m})}, \boldsymbol{h}^{(\bar{m})}, \tau))$ (no gradient; if $\tau{=}0$: $\boldsymbol{D}^{(\bar{m})} \leftarrow \mathrm{sg}(\boldsymbol{D}^{(m)})$)
  $\ell(\theta) \leftarrow \ell_{\text{fCRPS}}(\boldsymbol{x}^{t+\Delta t}, \boldsymbol{D}^{(m)}, \boldsymbol{D}^{(\bar{m})}; \theta)$ (Equation (9))
  $\theta \leftarrow \text{optimiser}(\ell(\theta), \theta)$ (gradient reaches $\boldsymbol{D}^{(m)}$ only)

  // Advance both members with the SDE Solver
  $\Delta\tau \leftarrow \min\big((1/S)\, U[1{-}j,\ 1{+}j],\ 1 - \tau\big)$
  for $m \in \{a, b\}$ **do**
    $\boldsymbol{X}_{+}^{(m)} \leftarrow \text{SDE Solver}\big(\boldsymbol{X}^{(m)},\ \boldsymbol{B}_\theta(\cdot, \boldsymbol{h}^{(m)}, \cdot),\ \tau,\ \Delta\tau\big)$ (drift function from Equation (6), value at τ should be cached from $\boldsymbol{D}^{(m)}$ and $\boldsymbol{D}^{(\bar{m})}$ to limit compute costs.)
  end for
  // Push to replay buffer and handle end of SDE trajectories at $\tau{=}1$
  if $\tau + \Delta\tau \geq 1$ **then**
    if $k + 1 < K_{\max}$ **and** $\boldsymbol{x}^{t+2\Delta t} \in \mathcal{D}$ **then**
      for $m \in \{a, b\}$ **do**
        $\boldsymbol{h}^{(m)} \leftarrow \text{slide}(\boldsymbol{h}^{(m)}, \boldsymbol{X}_{+}^{(m)})$ (generated state becomes next input)
      end for
      push $(\boldsymbol{X}_{+}^{(a)}, \boldsymbol{X}_{+}^{(b)}, \boldsymbol{h}^{(a)}, \boldsymbol{h}^{(b)},\ \tau{=}0,\ k{+}1,\ t{+}\Delta t)$ to $\mathcal{B}$
    end if
      (else the lineage terminates; $\mathcal{B}$ shrinks $\Rightarrow$ next step injects a fresh sample)
  else
    push $(\boldsymbol{X}_{+}^{(a)}, \boldsymbol{X}_{+}^{(b)}, \boldsymbol{h}^{(a)}, \boldsymbol{h}^{(b)},\ \tau + \Delta\tau,\ k,\ t)$ to $\mathcal{B}$
  end if
end for
Output: Finetuned denoiser $\boldsymbol{D}_\theta$