

Demystifying Data-Driven Probabilistic Medium-Range Weather Forecasting

Jean Kossaifi^{*†}, Nikola Kovachki^{*†}, Morteza Mardani*, Daniel Leibovici*, Suman Ravuri*, Ira Shokar*, Edoardo Calvello, Mohammad Shoib Abbas, Peter Harrington, Ashay Subramaniam, Noah Brenowitz, Boris Bonev, Wonmin Byeon, Karsten Kreis, Dale Durran, Arash Vahdat, Mike Pritchard and Jan Kautz
NVIDIA

Abstract

The recent revolution in data-driven methods for weather forecasting has lead to a fragmented landscape of complex, bespoke architectures and training strategies, obscuring the fundamental drivers of forecast accuracy. Here, we demonstrate that state-of-the-art probabilistic skill requires neither intricate architectural constraints nor specialized training heuristics. We introduce a scalable framework for learning multi-scale atmospheric dynamics by combining a directly downsampled latent space with a history-conditioned local projector that resolves high-resolution physics. We find that this simple design is robust to the choice of probabilistic estimator, seamlessly supporting stochastic interpolants, diffusion models, and CRPS-based ensemble training. Validated against the Integrated Forecasting System and the deep learning probabilistic model GenCast, our framework achieves statistically significant improvements on most of the variables. These results suggest that for medium-range prediction, scaling a unified, general-purpose model can be more effective than relying on domain-specific complexity or probabilistic estimation framework.

1. Introduction

Until recently, the consensus within the atmospheric science community was that purely data-driven approaches could never rival the fidelity of numerical weather prediction (NWP). The chaotic, multi-scale dynamics of the atmosphere were thought to demand explicit solvers based on fluid dynamics. However, the introduction of a neural-operator [1], and later graph-neural-network approaches [2, 3] for capturing these multi-scale dynamics, demonstrated that deep learning could not only capture global circulation patterns from historical data but do so with accuracy competing with the gold-standard Integrated Forecasting System (IFS) developed by the European Centre for Medium-Range Weather Forecasts (ECMWF). Since, data-driven deterministic models [1, 4, 5, 6, 7, 8] have reached parity with or surpassed traditional NWP.

Beyond matching the accuracy of traditional solvers, these models offer a fundamental computational advantage: they generate predictions several orders of magnitude faster, often requiring only a single GPU. This efficiency has democratized access to high-quality forecasting and opened the door to massive ensembles [9, 10]. Consequently, the frontier of research has now shifted from establishing feasibility to mastering probabilistic prediction [11, 12, 13, 14, 15], aiming to capture the uncertainty inherent in partially observed, chaotic systems.

However, these recent and rapid evolutions of the field have led to a fragmented landscape of architectures and training methodologies. Because weather data resides on a spherical manifold and involves distinct physical variables, researchers have heavily prioritized domain-specific inductive biases. Current approaches generally fall into three distinct design philosophies. First, the FourCastNet family [6, 14] and related architectures [16, 17, 18, 19, 20] leverage spherical harmonics and convolutions to strictly enforce geometric symmetries as inductive biases. Second, graph-based models [5, 11, 12, 13] encode the atmosphere on icosahedral meshes and rely on message-passing architectures to capture physical dynamics. A third category [4, 7, 21] adapts vision transformers (ViTs) [22] with bespoke tokenization strategies or vertical aggregation layers to handle the 3D structure of the atmosphere. While effective, these specialized architectural and training constraints often introduce significant engineering complexity and computational bottlenecks that hinder

^{*} Core contributors.

[†] Corresponding authors: {jkossaifi, nkovachki}@nvidia.com.

© 2026 NVIDIA. All rights reserved.

efficient scaling.

The challenge is compounded in the probabilistic setting where the design space is even more fragmented. To capture uncertainty, complex loss functions and sampling strategies have been proposed. Some approaches optimize continuous-ranked probability scores (CRPS) directly but can struggle to accurately predict spectral power at the shortest retained wavelengths [12, 13]. Others augment CRPS with spectral terms to improve sharpness [23, 14].

More recently, diffusion, flow-matching, or consistency models [11, 24, 25] have demonstrated high fidelity but suffer from slow inference times or limited resolution. Indeed, diffusion based approaches have especially attractive potential for controllability relevant to multi-task foundation modeling [26] beyond the prediction task. However, they have so far not achieved probabilistic skill close to state-of-the-art without sacrificing either time or spatial resolution – an open challenge.

We investigate the extent to which bespoke probabilistic estimation or architectural complexity are necessary to achieve state-of-the-art weather forecasts by pursuing a very different path: a streamlined framework that relies on standard, scalable components. Our framework leverages standard, scalable transformer architectures operating within a compressed latent space. Inspired by the success of latent diffusion in computer vision, our approach decouples the modeling of global dynamics from high-resolution synthesis. By stripping away intricate, domain-specific constraints in favor of a general-purpose, scalable backbone, we achieve state-of-the-art probabilistic performance with simplicity.

Our contributions are as follows:

- **A unified latent framework:** We find that the direct application of probabilistic methods such as diffusion or stochastic interpolants to high-resolution weather data is often computationally prohibitive or unstable. We propose the ATLAS framework: a standard transformer operating in a directly downsampled latent space, coupled with a history-conditioned local projector. We show that this streamlined framework effectively resolves high-resolution physics without the need for complex variational encodings or bespoke layers.
- **Method-agnostic robustness:** We demonstrate that this framework serves as a universal backbone for probabilistic forecasting. Unlike prior works that are tightly coupled to specific loss functions or training recipes, our framework achieves state-of-the-art results using three independent estimation methods, including stochastic interpolants, diffusion models, and CRPS-based ensemble training, suggesting potential to unify the fragmented modeling landscape.
- **State-of-the-art performance for a fully open model:** We validate our approach against the Integrated Forecasting System (IFS) and the strongest publicly available deep learning probabilistic model we are aware of that was trained on similar ERA5 data and can therefore be fairly compared – GenCast. Our method outperforms the IFS by a large margin and achieves statistically significant improvements over GenCast for most variables (Figure 1).
- This performance represents the most skillful attempt yet to train a diffusion-based approach for this task without sacrificing resolution.

2. ATLAS: Medium-Range Latent Probabilistic Weather Forecasting

In this section, we introduce ATLAS (Atmospheric Transformer in Latent Space), our method for medium-range probabilistic weather forecasting. We first introduce the problem formulation, our choice of latent space modeling and backbone architectures before introducing three probabilistic generative methods used to train and validate our approach.

2.1. Problem Formulation

We consider a stationary, discrete-time, \mathbb{R}^d -valued stochastic process $\{x_j\}_{j \in \mathbb{Z}}$ which represents the distribution of d measurements of the state of the atmosphere at equally spaced time intervals. Our task is to model the conditional distribution $\rho_c(x_{j+1}|x_j)$ from a single, finite, realization of $\{x_j\}$ as data. Note that by the stationarity assumption, the joint distributions $\rho(x_j, x_{j+1}) = \rho(x_k, x_{k+1})$ for any $j, k \in \mathbb{Z}$ hence ρ_c is independent of time.

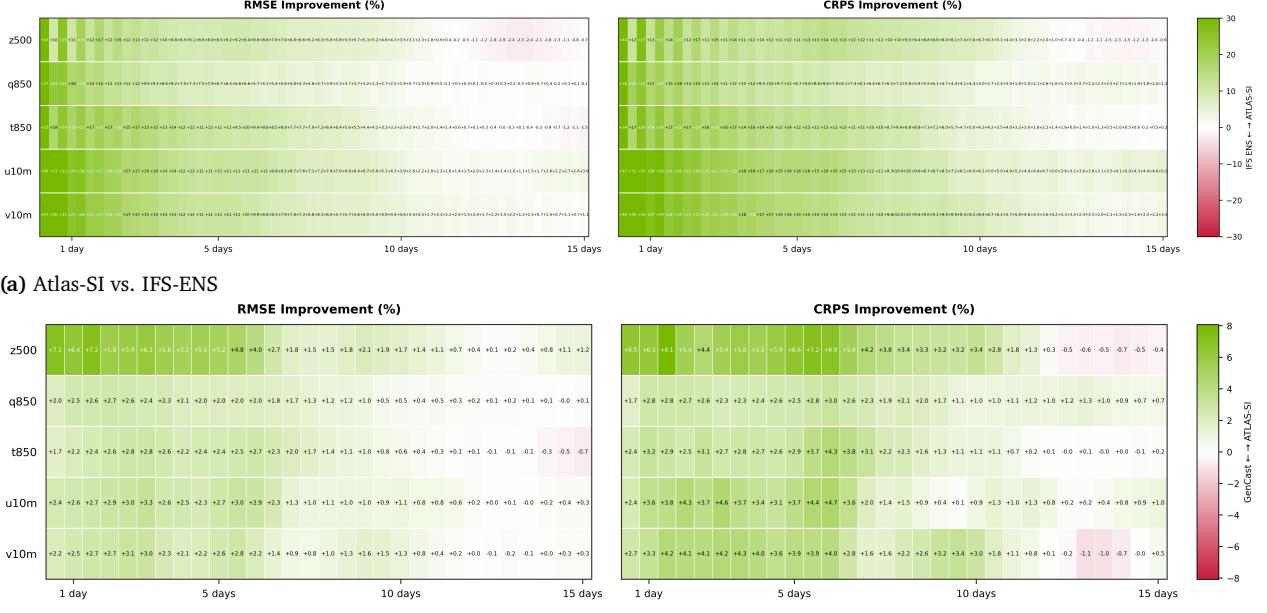


Figure 1: Scorecard comparison of ATLAS-SI vs. IFS-ENS (a) and GenCast (b) for a fifteen day forecast with 56 ensemble members averaged over ERA5 initial conditions in 2020. Left shows percent improvement in the RMSE of the ensemble mean and right in the ensemble CRPS.

This allows us to formulate the problem as finding a model for a single conditional distribution $\rho_c(x_1|x_0)$ given data in the form of $N \in \mathbb{N}$ realizations from the joint distribution $\rho(x_0, x_1)$. In particular, we assume to have data $\{(x_j^\dagger, x_{j+1}^\dagger)\}_{j=0}^{N-1}$ such that the pairs $(x_j^\dagger, x_{j+1}^\dagger) \sim \rho(x_0, x_1)$ are identically distributed. While we formulate our models within this framework for simplicity, we readily admit that the stationarity assumption may be violated for real weather data. This problem may be dealt with by appropriately conditioning on the sources of non-stationarity, e.g., a representation of the time of day and the day of the year.

In subsection 2.4, we formulate three approximate models for $\rho_c(x_1|x_0)$. Each method defines a conditional transport map which, given an initial condition from $\rho(x_0)$, pushes a known probability distribution to $\rho_c(x_1|x_0)$. Therefore sampling $\rho_c(x_1|x_0)$ can be done by first sampling the known distribution and then evaluating the transport map. In subsections 2.4.1 and 2.4.2, the transport map is defined implicitly as the flow map of a stochastic process with $\rho_c(x_1|x_0)$ as its terminal condition, while in subsection 2.4.3 it is defined directly. Each case requires parameterizing a map which takes as input x_0 as well as other case dependent inputs to produce a \mathbb{R}^d -valued output. In subsequent sections, we present a unified framework for parameterizing such maps as well as reducing the dimensionality of the original data that is employed in all methods outlined in subsection 2.4.

Given an initial sample $x_0 \sim \rho(x_0)$, we may use the models formulated in subsection 2.4 to generate an approximate sample $\hat{x}_1 \sim \rho_c(x_1|x_0)$ for the next time step. We may then view \hat{x}_1 as the initial sample and again use our approximate models to produce a sample for the next time step $\hat{x}_2 \sim \rho_c(x_1|\hat{x}_1)$. Iterating on this process, called unrolling, gives a forecast $\hat{x}_1, \hat{x}_2, \dots$ for the initial condition x_0 . Since \hat{x}_1 is only approximately distributed according to $\rho_c(x_1|x_0)$, using it as an initial sample for the next step introduces a distribution shift in the inputs of the transport maps of subsection 2.4. As more steps are unrolled, the distribution shift can become worse because errors may compound. We show that our architectures are robust to this distribution shift and can produce stable forecasts up to 15 days, amounting to 60 auto-regressive steps.

2.2. Latent Space Modeling

Latent space

We model a time series of re-assimilated observations of the atmosphere from ECMWF’s ERA5 dataset (see subsection 3.1 for details). Each data point consists of 75 variables each on a quarter degree 721×1440

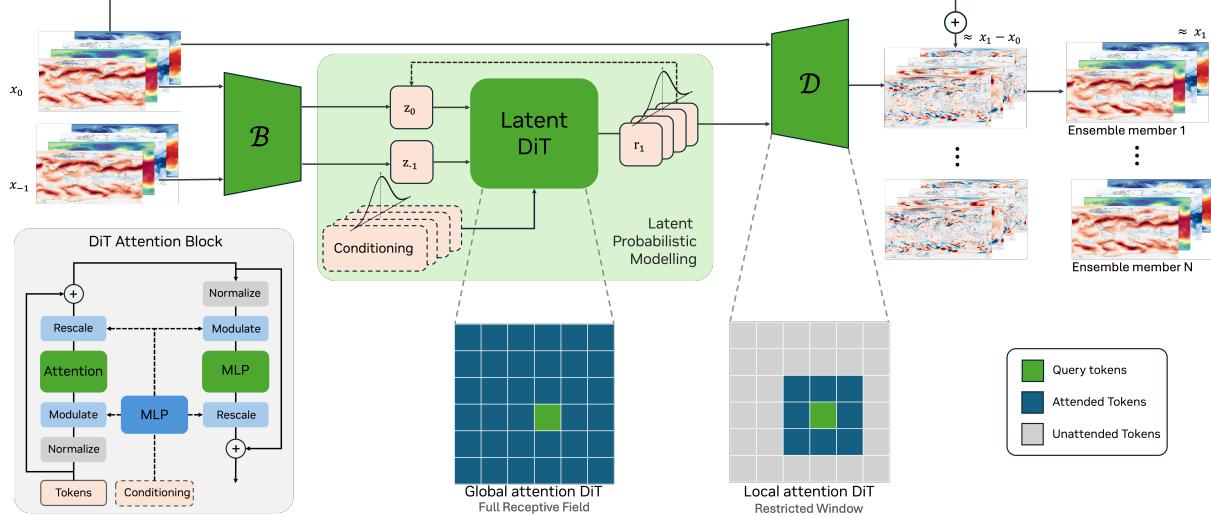


Figure 2: Overview of our approach: Atmospheric states x_0 and x_{-1} are encoded into a latent representation z_0 and z_{-1} , which is mapped, through a probabilistic model with a transformer backbone into a latent representation of residuals r_1 of the next time step. These residuals are deterministically decoded back into original space with a transformer-based decoder which is also conditioned on the high resolution initial state.

equiangular grid, making $d = 77,868,000$. Handling such high-dimensional data directly is computationally challenging for modern deep learning architectures [27]. We therefore adopt a latent space approach that has recently shown success for computer vision and natural language processing tasks [28, 29]. In particular, we encode each data point to a lower dimensional space, perform probabilistic inference in that space, and decode the output back to the original space. In computer vision, this is normally done by first training an encoder-decoder architecture and then training probabilistic models on the encoded data. We take a simpler approach and instead downsample the data using bilinear interpolation and then train a decoder-only model to upsample. We find this approach to produce considerably better behaved latent spaces with favorable spectral properties, leading to probabilistic forecast models that remain stable in rollout. Standard VAEs mix all channels (here representing physical variables) in the encoder, producing latent features that aggregate heterogeneous variables into a common channel space. However initial experiments indicated that such mixing tends to: (i) generate latent spectra with a long, flat high-frequency tail (a manifestation of spectral bias when aggressively compressing complex multi-channel fields [30]), and (ii) ignore the underlying temporal structure of the data, so that temporally adjacent fields can be mapped far apart in latent space [31, 32, 33]. By using bilinear interpolation, we remove only the small-scale physics from the data, which are usually only predictable at short time scales, but preserve other physical structures.

Furthermore since memory and computational resources that are normally reserved for the encoder can now be put into the decoder, we find that this method achieves significantly better reconstruction performance. We choose a $16\times$ compression scheme by bilinearly interpolating each variable to a one degree 181×360 equiangular grid. Since we choose to make predictions at six hour intervals, this choice is physically justified because, at that time scale, spatial scales less than approximately seventy kilometers are deterministically unpredictable [34]. We find no degradation in predictive performance when upsampling with our decoder model since its reconstruction errors tend to be an order of magnitude lower than those of the predictive model. We refer to section 5 for ablations of various encoding strategies and comparisons against encoder-decoder models.

Residual prediction and history

Empirical tests also uncovered better performance when predicting temporal residuals of states instead of the states directly. We also found that including one additional historical state improves the robustness of the model, improving stability at longer lead times.

Let \mathcal{B} denote the bilinear downsampling operator. We will write $z_j = \mathcal{B}(x_j)$ for the latent state and

$r_{j+1} = \mathcal{B}(x_{j+1} - x_j)$ for the latent residual. Our aim then is to approximate the distribution $\rho_c(r_{j+1}|z_j, z_{j-1})$. We will assume that the application of \mathcal{B} preserves stationarity, and therefore that this is equivalent to the distribution $\rho_c(r_1|z_0, z_{-1})$ for all $j \in \mathbb{Z}$. Let \mathcal{D} denote the decoder model that approximately inverts \mathcal{B} . It then follows that if $y \sim \rho_c(r_1|z_0, z_{-1})$, $\mathcal{D}(y) + x_0$ is approximately distributed according to $\rho_c(x_1|x_0, x_{-1})$. We therefore train a decoder model that upsamples residuals so that $\mathcal{D}(r_1) \approx x_1 - x_0$. We have found that including the initial state as an input to this model significantly improves reconstruction performance, that is, $\mathcal{D}(r_1, x_0) \approx x_1 - x_0$. This makes \mathcal{D} similar to a deterministic model which predicts residuals but also has access to a downsampled version of the truth. This mix of data allows the model to more easily capture high-resolution details and the necessary physical dynamics from its input. Figure 2 depicts a graphical summary of our approach.

2.3. Transformer Architectures as the backbone of ATLAS

As a base model, we adapt the Diffusion Transformer (DiT) [35] originally popularized for image generation. We tailor the specifics of the DiT to construct the decoder architecture parameterizing \mathcal{D} and the predictive architecture used for modeling $\rho_c(r_1|z_0, z_{-1})$. In latent image and video diffusion models, it is common practice to use a convolutional U-Net decoder [36] and a transformer for the generative model [27, 28, 37]. We find that using a transformer-based decoder significantly improves our reconstruction results with errors typically an order of magnitude lower than those of the predictive results. This improvement makes our latent space approach possible as we find no loss in performance when going from the latent to the original space.

Decoder architecture

Our decoder model \mathcal{D} takes as input the latent residual r_1 which is a 181×360 spatial field with 75 channels and the full resolution initial condition x_0 which is a 721×1440 spatial field with 79 channels (see subsection 3.1 for details). It then outputs an approximation to the residual $x_1 - x_0$ which is again a 721×1440 spatial field with 75 channels. In order to align the two spatial fields, we first patch x_0 into 181×360 tokens by applying a learned strided convolution with a 1×1 kernel, expanding the channels to an embedding dimension $e_o \in \mathbb{N}$. We similarly apply a 1×1 non-strided convolution to the residual field r_1 , expanding its channels to $e_1 \in \mathbb{N}$. We then concatenate the two fields into a single 181×360 spatial field with $e = e_o + e_1$ channels. A standard sine-cosine positional encoding is added to this field and $l \in \mathbb{N}$ DiT blocks are applied to it. Each DiT block is modified to use local attention [38] in a 3×3 window and minimal spherically consistent padding applied before the attention operation to ensure each pixel in the true field has a 3×3 context window around it. Since the task of the decoder is spatially local, we have found local attention to be incredibly effective while being an order of magnitude faster than its global counterpart on the long sequence length $181 \cdot 360 = 65,160$. After applying the last DiT block, a local linear layer is applied transforming the $e \times 181 \times 360$ field to a $75 \times 721 \times 1440$ field. Since the original DiT was designed for use in diffusion models, each DiT block takes as input an extra time-conditioning parameter t which internally modulates the scale and shift of the block's output. We keep this extra conditioning, simply fixing it to $t = 1$ for any input as we have found that this performs better than removing it.

Predictive architecture

Similar to the decoder architecture, the predictive architecture used to parameterize the maps \hat{b} , \hat{s} , or \hat{f} is based on a DiT with various number of input fields. For \hat{b} (or \hat{s}) the inputs are the noisy field X_t^{SI} (or X_t^{DM}) and the two historical states z_0, z_{-1} , all of which are $75 \times 181 \times 360$ spatial fields. We use a strided convolution with a 1×1 kernel to patch the first input field to a dimension $e_1 \times p_1 \times p_2$ and a similar convolution for the concatenation of the historical states (z_0, z_{-1}) resulting in a $e_2 \times p_1 \times p_2$ field. Subsection 2.4 provides further details.

Experimentation indicated that smaller patch sizes, resulting in larger p_1, p_2 , yield a consistently better single time step predictive performance at the expense of a higher computational and memory cost. However, at very small patch sizes, symptoms of overfitting emerge, and predictions become unstable during rollout. We find that 2×3 patches, which make $p_1 = 91$ and $p_2 = 120$, to be the best compromise, resulting in accurate and stable models. For equiangular grids, this patch size is known to give quasi-isotropic patches and has some precedent in numerical methods for climate modeling [39]. Once patched, the inputs are concatenated into a single representation with embedding dimension $e = e_1 + e_2$ and sequence length $p_1 \cdot p_2 = 10,920$. A standard sine-cosine embedding is added to this representation and $l \in \mathbb{N}$ DiT blocks are applied each conditioned on

the time input $t \in \mathbb{R}$. In this architecture, we use global attention as we find that it significantly improves the stability of the model compared to its local counterpart. This may be due to potential long-range correlations in the dynamics of the atmosphere. The final state is again obtained by a local linear layer projecting to a $75 \times 181 \times 360$ field. In the case of \hat{f} , the inputs are the fields z_0, z_{-1} and Gaussian vector $\xi \sim N(0, I_p)$. The spatial fields are treated as before while the Gaussian vector is used as extra conditioning to the DiT blocks similar to the architecture in [12].

2.4. Three Approaches to Probabilistic Modeling in Latent Space

In the following three subsections, we describe three probabilistic methods that, once trained, allow us to approximately sample the latent conditional distribution $\rho_c(r_1|z_0, z_{-1})$. Combined with our decoder, this gives an approximate method for sampling $\rho_c(x_1|x_0)$ as described in subsection 2.2. For notational simplicity, we will omit z_{-1} from our formulation and simply note that it is added as an additional input to any parameterized map. Furthermore, we denote by $z_j^\dagger = \mathcal{B}(x_j^\dagger)$ and $r_{j+1}^\dagger = \mathcal{B}(x_{j+1}^\dagger - x_j^\dagger)$, for all $0 \leq j \leq N-1$, the latent encodings of our dataset. For simplicity, we continue to denote by d the dimensionality of our data, noting that below this now represents the dimensionality of the latent encoding and not the original data.

2.4.1. Stochastic Interpolants (SI)

Our first modeling approach is based on the stochastic interpolants framework [40] and, in particular, the time-series formulation in [41]. Define the stochastic process

$$I_t = \alpha(t)z_0 + \beta(t)r_1 + \sigma(t)W_t, \quad 0 \leq t \leq 1,$$

where $(z_0, r_1) \sim \rho(z_0, r_1)$, W_t is a \mathbb{R}^d -valued Wiener process independent of (z_0, r_1) , and $\alpha, \beta, \sigma \in C^1([0, 1]; \mathbb{R}_{\geq 0})$ satisfy the boundary conditions $\alpha(0) = 1$, $\alpha(1) = 0$, $\beta(0) = 0$, $\beta(1) = 1$, $\sigma(0) = \sigma(1) = 0$. We therefore have that $I_0 = z_0$, $I_1 = r_1$ and, in particular, $I_t|z_0$ is a bridge between δ_{z_0} and $\rho_c(r_1|z_0)$. It is shown in [41, Theorem 3.1], that solutions of the stochastic differential equation (SDE)

$$dX_t^{\text{SI}} = b(X_t^{\text{SI}}, z_0, t)dt + \sigma(t)dW_t, \quad X_0^{\text{SI}} = z_0 \quad (1)$$

satisfy $\text{Law}(X_t^{\text{SI}}) = \text{Law}(I_t|z_0)$ for all $t \in [0, 1]$ where $b : \mathbb{R}^d \times \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}$ is the minimizer of

$$\mathcal{L}_{\text{SI}} = \int_0^1 \mathbb{E}[|b(I_t, z_0, t) - \dot{\alpha}(t)z_0 - \dot{\beta}(t)r_1 - \dot{\sigma}(t)W_t|^2]dt \quad (2)$$

with the expectation taken over the joint distribution (z_0, r_1, W_t) . In particular, we note that $X_1^{\text{SI}} \sim \rho_c(r_1|z_0)$, hence solving (1), allows us to sample $\rho_c(r_1|z_0)$.

By discretizing the expectation in (2) with data samples, we may find an approximate minimizer of b . In particular, we define \hat{b} to be the minimizer (over the parametric family defined in subsection 2.3) of

$$\hat{\mathcal{L}}_{\text{SI}} = \frac{1}{N} \sum_{j=0}^{N-1} \mathbb{E}_{t \sim U(0,1)} \mathbb{E}_{\xi \sim N(0,I)} [|\hat{b}(I_{t,j}^\dagger, z_j^\dagger, t) - \dot{\alpha}(t)z_j^\dagger - \dot{\beta}(t)r_{j+1}^\dagger - \dot{\sigma}(t)\sqrt{t}\xi|^2]$$

where $I_{t,j}^\dagger = \alpha(t)z_j^\dagger + \beta(t)r_{j+1}^\dagger + \sigma(t)\sqrt{t}\xi$. In practice, similarly to [42], we re-parametrize the loss so that inputs and outputs to \hat{b} have approximately zero mean and unit variance at all time steps $t \in [0, 1]$ and pick a linear schedule for α, β, σ . Using this minimizer in (1), gives an approximate model of $\rho_c(r_1|z_0)$. That is, we define

$$d\hat{X}_t^{\text{SI}} = \hat{b}(\hat{X}_t^{\text{SI}}, z_0, t)dt + \sigma(t)dW_t, \quad \hat{X}_0^{\text{SI}} = z_0 \quad (3)$$

and expect that $\rho(\hat{X}_1^{\text{SI}}) \approx \rho_c(r_1|z_0)$ since $\hat{b} \approx b$. Numerically, we discretize (3) with a first-order stochastic Runge-Kutta scheme [43].

2.4.2. Diffusion Models

Our second modeling approach is based on the diffusion models framework [44] and, in particular, the formulation in [42]. Consider the system of SDE(s)

$$\begin{aligned} dX_t^{\text{F},0} &= 0, & X_0^{\text{F},0} &= z_0, \\ dX_t^{\text{F},1} &= \sqrt{2\sigma(t)\dot{\sigma}(t)}dW_t, & X_0^{\text{F},1} &= r_1, \end{aligned}$$

for $t \in [0, \infty)$ where $(z_0, r_1) \sim \rho(z_0, r_1)$, $\sigma \in C^1([0, \infty); \mathbb{R}_{\geq 0})$ monotonically increasing with $\sigma(0) = 0$, and W_t is a \mathbb{R}^d -valued Wiener process independent of (z_0, r_1) . Let $p : \mathbb{R}^d \times \mathbb{R}^d \times [0, \infty) \rightarrow \mathbb{R}$ denote the density of $(X_t^{F,1}, X_t^{F,0})$. We now consider the reverse time SDE,

$$dX_t^{\text{DM}} = -2\sigma(t)\dot{\sigma}(t)\nabla \log p(X_t^{\text{DM}}, z_0, t)dt + \sqrt{2\sigma(t)\dot{\sigma}(t)}d\bar{W}_t, \quad X_T^{\text{DM}} = X_T^{F,1}, \quad (4)$$

for some $T > 0$ where ∇ denotes the gradient in the first component and \bar{W}_t is a reverse time, \mathbb{R}^d -valued Wiener process independent of (z_0, r_1, W_t) . It can be shown that $\text{Law}(X_t^{\text{DM}}) = \text{Law}(X_t^{F,1})$ for $t \in [0, T]$ and, in particular, $X_0^{\text{DM}} \sim \rho_c(r_1 | z_0)$ hence solving (4) allows us to sample $\rho_c(r_1 | z_0)$ [44].

To that end, we estimate $\nabla \log p$ from data. It is shown in [45, Theorem 1] that $\nabla \log p$ is the minimizer of

$$\mathcal{L}_{\text{DM}} = \int_0^T \mathbb{E}[|\nabla \log p(X_t^{F,1}, X_t^{F,0}, t) - \sigma(t)^{-2}(X_t^{F,1} - r_1)|^2]dt \quad (5)$$

where the expectation is taken over $(X_t^{F,0}, X_t^{F,1})$. We therefore define \hat{s} to be the minimizer of

$$\hat{\mathcal{L}}_{\text{DM}} = \frac{1}{N} \sum_{j=0}^{N-1} \mathbb{E}_{t \sim U(0, T)} \mathbb{E}_{\xi \sim N(0, I)} |\hat{s}(r_{j+1}^\dagger + \sigma(t)\xi, z_j^\dagger, t) - \sigma(t)^{-1}\xi|^2.$$

Using this minimizer in (4) and approximating the terminal condition gives an approximate model of $\rho_c(r_1 | z_0)$. That is, we define,

$$d\hat{X}_t^{\text{DM}} = -2\sigma(t)\dot{\sigma}(t)\hat{s}(\hat{X}_t^{\text{DM}}, z_0, t)dt + \sqrt{2\dot{\sigma}(t)\sigma(t)}d\bar{W}_t, \quad \hat{X}_T^{\text{DM}} = \xi_T \quad (6)$$

where $\xi_T \sim N(0, (1 + \sigma(T)^2)I)$. Then we expect that $\rho(\hat{X}_0^{\text{DM}}) \approx \rho_c(r_1 | z_0)$ since $\hat{s} \approx \nabla \log p$ and $\rho(\xi_T) \approx \rho(X_T^{F,1})$ for T large enough. Numerically, we discretize (6) with the predictor-corrector method introduced in [42].

2.4.3. CRPS-based Models

Our third modeling approach learns a direct generative map $\hat{f} : \mathbb{R}^d \times \mathbb{R}^p \rightarrow \mathbb{R}^d$ such that, for a latent state z_0 and noise $\xi \sim N(0, I_p)$, the output $\hat{f}(z_0, \xi)$ approximates a sample from the conditional distribution $\rho_c(r_1 | z_0)$. Training is performed by minimising a generalised Continuous Ranked Probability Score (CRPS) [46, 47] using the ℓ_1 -norm, chosen for robustness in weather forecasting [12, 13], despite not being strictly proper for general distributions [48, 49].

For a given (z_0, r_1) pair and independent noise samples $\xi, \xi' \sim N(0, I_p)$, the CRPS objective is

$$\mathcal{L}_{\text{CRPS}}(f) = \mathbb{E}\left[|f(z_0, \xi) - r_1| - \frac{1}{2}|f(z_0, \xi) - f(z_0, \xi')|\right], \quad (7)$$

where the first term measures the distance to the target and the second penalizes excessive ensemble spread.

This CRPS objective compares predictive ensembles to observations only through marginal distributions and does not explicitly enforce the joint spatial structure. While the scalar CRPS is strictly proper, this property does not extend to aggregated CRPS over spatial or multivariate dimensions, allowing ensembles to be point-wise accurate yet inconsistent jointly. This manifests in two ways: CRPS-based models do not guarantee physically meaningful spatial correlations, and empirically they exhibit spectral bias, capturing large-scale modes while under-representing high-frequency energy [23].

To address this, we introduce a spectral regularization term. Let \mathcal{S} denote the spherical harmonic transform, implemented using `torch-harmonics` [6]. For a spatial field $x(\theta, \phi)$ on the sphere S^2 , \mathcal{S} computes the spectral coefficients via the projection:

$$[\mathcal{S}(x)]_{l,m} = \int_{S^2} x(\theta, \phi) Y_{l,m}^*(\theta, \phi) d\Omega, \quad (8)$$

where $Y_{l,m}$ are the orthonormal spherical harmonic basis functions, $d\Omega = \sin \theta d\theta d\phi$ is the area element, and $(\cdot)^*$ denotes the complex conjugate. We compute the CRPS on the magnitudes of the spectral coefficients $|\mathcal{S}(\cdot)|$, ensuring phase invariance.

Lang et al. [12] observed that the fair CRPS (an unbiased finite-ensemble estimator) can exhibit a degeneracy in which ensemble variability is under-constrained when most members exactly match the observation; to mitigate this, they propose using a mixture of biased and fair CRPS estimators. Similarly, Alet et al. [13] restrict the latent noise dimensionality to constrain stochastic variance and promote coherent global structures. In contrast, we find that spectral CRPS regularization not only stabilizes training with high-dimensional noise with the standard two-sample CRPS estimator but also allows full expressive stochastic conditioning without biased-fair mixtures or low-dimensional noise constraints.

During optimization, we discretize the outer expectation by sampling a training point z_j^\dagger and approximate the inner expectations using two independent noise samples:

$$\begin{aligned}\hat{\mathcal{L}}(f) = & \left[|f(z_j^\dagger, \xi_1) - r_{j+1}^\dagger| + |f(z_j^\dagger, \xi_2) - r_{j+1}^\dagger| - |f(z_j^\dagger, \xi_1) - f(z_j^\dagger, \xi_2)| \right] \\ & + \lambda_{\text{spec}} \left[\|\mathcal{S}(f(z_j^\dagger, \xi_1))| - |\mathcal{S}(r_{j+1}^\dagger)\| + \|\mathcal{S}(f(z_j^\dagger, \xi_2))| - |\mathcal{S}(r_{j+1}^\dagger)\| \right. \\ & \left. - \|\mathcal{S}(f(z_j^\dagger, \xi_1))| - |\mathcal{S}(f(z_j^\dagger, \xi_2))\| \right],\end{aligned}$$

with $\xi_1, \xi_2 \sim N(0, I_p)$ i.i.d and λ_{spec} denotes a weighting factor. Given the minimizer \hat{f} , we expect that for a latent state z_0 and $\xi \sim N(0, I_p)$, $\hat{f}(z_0, \xi) \sim \rho_c(r_1|z_0)$, i.e., a single draw of ξ passed through \hat{f} yields an approximate sample from the conditional distribution.

Together our latent space encoding strategy described in subsection 2.2, our transformer-based decoder and predictive architectures described in subsection 2.3, and the probabilistic models described in subsection 2.4 constitute our proposed model ATLAS. The full method is illustrated in Figure 2.

3. Experimental setting

In this section, we detail the dataset and training parameters that are common across all our models. We further delineate our evaluation protocol and the metrics used to obtain all quantitative results.

3.1. Training and evaluation

We use standard methods to compare the performance of the models on key metrics/variables with each other and with our chosen baselines.

Dataset

We use a subset of ECMWF'S ERA5 dataset at the original 721×1440 resolution given at 6 hour intervals for years from 1980 to 2019 for training, and 2020 for testing. We include seven surface variables and five atmospheric variables each at thirteen different pressure levels (see Table 1). We include two additional variables: sea surface temperature (`sst`) and total precipitation (`tp`), for a total of 75 input and output variables. At any land location, we assign zero values to the `sst` field.

Training procedure

Prior to training, all data is normalized with statistics computed from the training years. In particular, we compute the mean and standard deviation of all pixel values separately for each field. We compute separate statistics for input fields and the residual fields. We then subtract the mean and divide by the standard deviation, so that each field follows an approximately standard Gaussian distribution. For evaluation, we apply the inverse of this normalization to our predictions and compute all metrics against the unnormalized dataset.

Each model is trained by minimizing its respective loss function detailed in subsection 2.4 by elastic averaged stochastic gradient decent with a total batch size of 32 spread over 32 80GB A100 or H100 GPUs [50]. We use the StableAdamW optimizer with default settings from the `optimi` library. We use the learning rate 1.28×10^{-4} with a linear ramp-up schedule for the first 2,000 steps followed by cosine decay for 100,000 steps [51], which we call a *cycle*. The learning rate is then reset to $0.8 \times$ of its original value and again cosine decayed for 100,000 steps. For the evaluated model, ATLAS-SI and ATLAS-EDM are trained for 3 cycles, while ATLAS-CRPS is trained for 6 cycles.

Channel	Description	ECMWF ID
Surface variables		
10u	10 meter u -wind component	165
10v	10 meter v -wind component	166
100u	100 meter u -wind component	228246
100v	100 meter v -wind component	228247
t2m	2 meter temperature	167
msl	Mean sea level pressure	151
tcwv	Total column vertically-integrated water vapor	137
Atmospheric variables at pressure level p indicated by -- in hPa		
z--	Geopotential	129
t--	Temperature	130
u--	u component of the wind	131
v--	v component of the wind	132
q--	Specific humidity	133

Table 1: Surface and atmospheric variables predicted by ATLAS. Detailed specifications of each variable can be accessed at <https://apps.ecmwf.int/codes/grib/param-db>.

All predictive models use the same decoder, which is also trained with the above settings for 3 cycles, minimizing the ℓ_1 -norm between its output and the residual field $x_1 - x_0$. To resolve potential non-stationarity in the data, our decoder also takes as input the cosine-zenith angle (date dependent), the surface geopotential, the land-sea mask, and the sst mask, all normalized in the range $[-1, 1]$, as additional fields appended to x_0 , giving a total of 79 input variables.

Model hyperparameters

Our three models — ATLAS-SI, ATLAS-EDM, and ATLAS-CRPS — use the same DiT predictive architecture described in subsection 2.3. We choose the embedding dimension of the current state to be $e_0 = 2496$ and the embedding dimension of the history to be $e_1 = 832$ for a total embedding dimension $e = 3,328$. We use $l = 12$ DiT blocks with 13 heads in the attention layer, resulting in an embedding dimension of 256 per head. For the SI and EDM versions, this results in a 2.4B parameter architecture. For the CRPS model, the noise dimension is chosen to be the same as the embedding dimension $p = 3,328$. This noise is first processed by a MLP and then passed to each DiT block as extra conditioning where it is processed by modulation layers. These extra layers result in a 3.3B parameter architecture. Our decoder model, common to all three configurations, has an embedding dimension for the high resolution state $e_0 = 1,728$ and an embedding dimension for the encoded residual $e_1 = 768$ for a total embedding dimension $e = 2,496$. We use $l = 16$ DiT blocks with 12 heads in the local attention layer, resulting in an embedding dimension of 208 per head. This results in a 1.8B parameter architecture.

Evaluation procedure

Evaluation is focused on headline probabilistic score metrics aggregated globally for the validation set (year 2020), and compared against a deterministic physics-based ensemble baseline from the ECMWF IFS ENS at 0.25 degree horizontal resolution, as well as against a strong data-driven baseline, GenCast.

Our discussion will especially focus on the skill in predicting the 500 hPa geopotential height ($z500$) as a key tracer of midlatitude dynamics. We also consider other important variables: 850 hPa temperature ($t850$), the total column water vapor ($tcwv$), 850 hPa humidity ($q850$), and the 10 meter wind velocities ($u10m$ and $v10m$). Our evaluation protocol uses statistics aggregated across forecasts launched on 28 common initialization dates spread throughout the year 2020, namely, 00:00 UTC on the 2nd and 16th of each month, plus the 8th for January–April. This will be shown to yield more than enough statistical power for validation against our baseline. Our models were evaluated at 6-hour intervals with rollouts extending up to 15 days. Each evaluation was conducted by simultaneously simulating 56 ensemble members and, at each forecast step, computing

the ensemble mean Root Mean Squared Error (RMSE), Continuous Ranked Probability Score (CRPS), and Spread–Skill Ratio (SSR). For the GenCast baseline, we unroll the model from the publicly available checkpoint (quarter degree, non-operational) using the same procedure as ATLAS but at 12 hour time steps. This setup was chosen to ensure a fair and consistent comparison with ATLAS.

We find that on a single A100 GPU, generating a single time-step prediction (12 hours for GenCast and 6 hours for ATLAS) takes around 140s for GenCast, 94s for ATLAS-SI, 88s for ATLAS-EDM, and 3.3s for ATLAS-CRPS.

3.2. Metrics

We evaluate our model on a basket of metrics generally used to assess probabilistic medium-range models. We evaluate all metrics on a 721×1440 equiangular grid and report results by variable, level, and lead time. We use the following notation for the rest of this section:

- Locations are indexed by p and represent a particular latitude and longitude.
- w_p is the area weight for that particular location and is proportional to $\sin(\text{lat}_u) - \sin(\text{lat}_l)$, where lat_u and lat_l denote the upper and lower bounds of the latitude at point p , respectively. Area weights are normalized such that $\sum_p w_p = 1$.
- We denote the ground truth at a particular initialization time t and location p as $x_{t,p}$. The number of initialization times is T .
- The prediction of $x_{t,p}$ with ensemble member m is $\hat{x}_{t,p}^{(m)}$. We define the mean prediction across M ensemble members $\bar{x}_{t,p}^{(m)} \equiv \frac{1}{M} \sum_m \hat{x}_{t,p}^{(m)}$.
- Due to the complexity of notation, variables, levels, and lead time are not indexed as above and are implied.

Our choice to evaluate ensemble members across 56 ensemble members could be viewed as non-standard given a frequent choice of 50. However this modest change is not material to our intercomparison since we also report results on baselines with 56 members, such that any comparisons between the two models are valid.

Continuous-Ranked Probability Score

To assess pointwise accuracy of the models, we evaluate on continuous-ranked probability score (CRPS) and ensemble mean RMSE. We use the “fair” version of CRPS [52], averaged across the above mentioned 28 initialization times. As the fair version is an unbiased estimator of CRPS, the average scores converge to the same value in the infinite sample limit regardless of ensemble size. The CRPS for a particular variable, level, and lead time is:

$$\text{CRPS} = \frac{1}{T} \sum_t \sum_p w_p \left(\frac{1}{M} \sum_m |\hat{x}_{t,p}^{(m)} - x_{t,p}| - \frac{1}{M(M-1)} \sum_{m \neq m'} |\hat{x}_{t,p}^{(m)} - \hat{x}_{t,p}^{(m')}| \right).$$

Ensemble-Mean RMSE

We further report the ensemble mean RMSE. Since we report the standard version, which is a biased estimate of the RMSE, there is a small bias due to the difference in ensemble member size. That said, the reported bias is lower than for a 50-member ensemble.

$$\text{ERMSE} = \sqrt{\frac{1}{T} \sum_t \sum_p w_p \left(\frac{1}{M} \sum_m (\bar{x}_{t,p}^{(m)} - x_{t,p})^2 \right)}.$$

Spread-Skill Ratio

To assess calibration, we report the spread-skill ratio (SSR) [53], where

$$\text{SSR} = \sqrt{\frac{M+1}{M}} \frac{\text{Spread}}{\text{ERMSE}}, \quad \text{and} \quad \text{Spread} = \sqrt{\frac{1}{T} \sum_t \sum_p w_p \left(\frac{1}{M-1} \sum_m (\hat{x}_{t,p}^{(m)} - \bar{x}_{t,p}^{(m)})^2 \right)}.$$

Under the assumption that ensemble members and the ground truth are exchangeable, a necessary condition is that the ensemble mean RMSE equals its spread. A properly calibrated model has a SSR of 1, while below and above that number represents under- and over-dispersion, respectively.

Statistical significance: Paired t-test

We apply a paired t-test to determine where differences between ATLAS skill are statistically detectable relative to the GenCast ensemble baseline, by computing p-values across initialization dates for each lead time. This approach recognizes the influence of sampling error and helps isolates genuine improvements in forecast skill from transient fluctuations in skill metrics from internal variability.

Let τ denote the forecast lead time. For each model m (Atlas-SI, Atlas-EDM, Atlas-CRPS, GenCast), let $S_m(t, \tau)$ denote the verification score (ERMSE or CRPS) evaluated at lead time τ . To compare two models A and B , we form paired score differences across identical initialization dates,

$$d_t(\tau) = S_A(t, \tau) - S_B(t, \tau).$$

For each lead time τ , we assess whether the mean score difference is statistically significant by testing the null hypothesis $H_0 : \mathbb{E}[d_t(\tau)] = 0$, using a paired t -test. The test statistic is given by

$$t(\tau) = \frac{\bar{d}(\tau)}{s_d(\tau)/\sqrt{T}},$$

where the sample mean and standard deviation of the paired differences are

$$\bar{d}(\tau) = \frac{1}{T} \sum_{t=1}^T d_t(\tau), \quad s_d(\tau) = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (d_t(\tau) - \bar{d}(\tau))^2}.$$

Under the null hypothesis, the statistic $t(\tau)$ follows a Student- t distribution with $T - 1$ degrees of freedom. Statistical significance is quantified using the corresponding two-sided p -value,

$$p(\tau) = 2 [1 - F_{t_{T-1}}(|t(\tau)|)],$$

where $F_{t_{T-1}}(\cdot)$ denotes the cumulative distribution function of the Student- t distribution. We say that a result is statistically significant if $p(\tau) < 0.05$.

4. Results

We perform a comprehensive quantitative and qualitative evaluation of ATLAS, demonstrating that the model family performs well on standard statistical metrics and extreme events.

The section begins with a quantitative evaluation of the ATLAS models, comparing them with strong data-driven and numerical models and to each other. Next, we perform a case study and highlight how the model can generate accurate and physically realistic predictions for meteorologically important events. Finally, this section concludes with a quantitative and qualitative evaluation on tropical cyclone tracking.

4.1. Quantitative Evaluation

In this section, we benchmark the ATLAS family of models against the ECMWF's Integrated Forecasting System and GenCast.

Comparison with Integrated Forecasting System

All three variants of ATLAS produce compelling skill gains relative to the IFS-ENS physics baseline, consistent with trends seen in other state-of-the-art data-driven approaches. Figure 3 demonstrates that these improvements are substantial at early lead times, particularly for surface variables; zonal and meridional surface

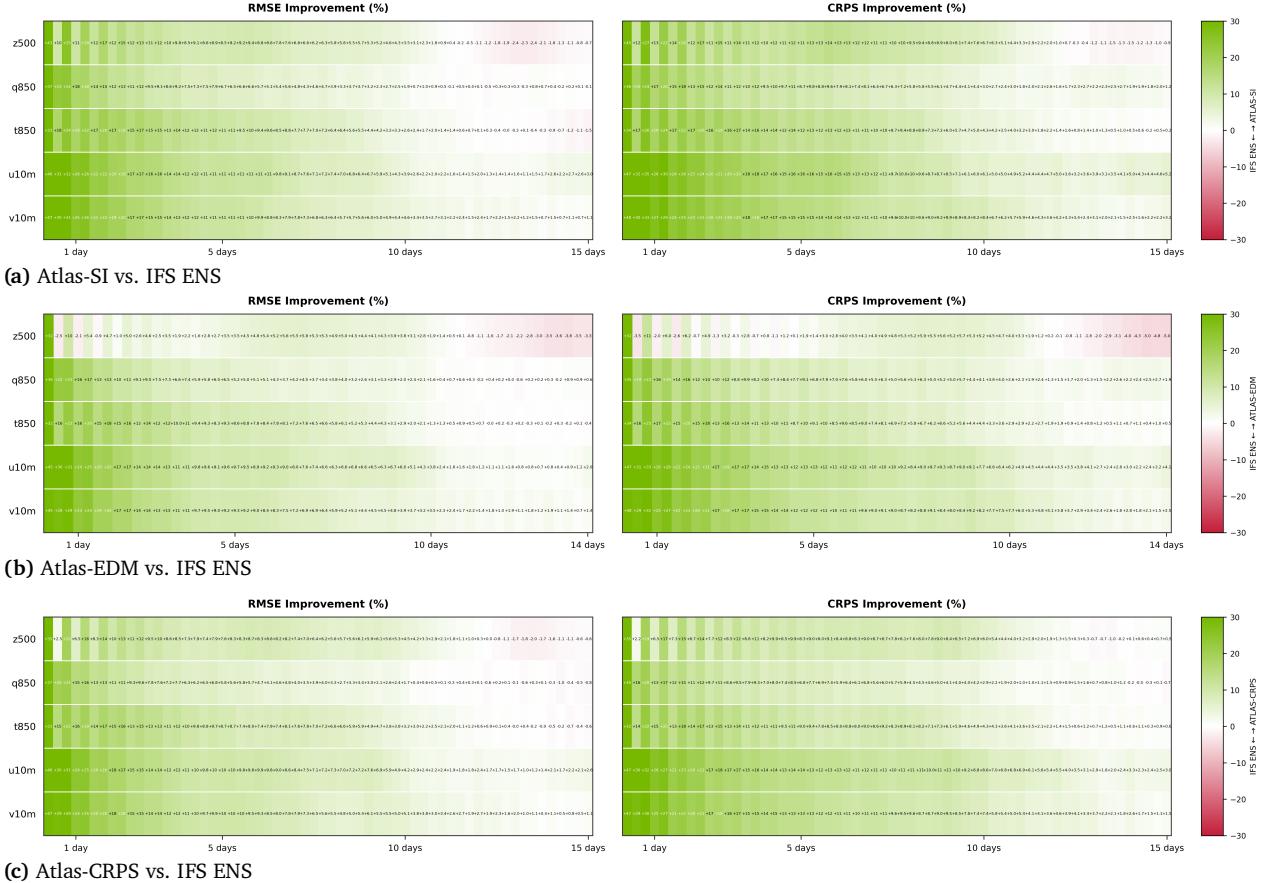


Figure 3: Heatmaps comparing ATLAS and IFS across RMSE and CRPS, for all three variants, SI, EDM and CRPS.

winds ($u10m$, $v10m$) exhibit the strongest relative performance, with CRPS improvements peaking near 50% at initialization and remaining above 10% throughout the first 6 days.

All three ATLAS variants maintain this performance advantage throughout the full 15-day forecast window for the majority of state variables, including specific humidity ($q850$), temperature ($t850$), and surface winds, where clear separation from the baseline persists even at two weeks. 500 hPa geopotential ($z500$) is an exception to this trend; here the skill advantage converges toward the baseline after approximately 12 days, whereas other atmospheric and surface fields retain improved skill over IFS-ENS through to day 15.

Comparison with GenCast

ATLAS also demonstrates superior skill against GenCast, a state-of-the-art open data-driven baseline. Figure 4 shows the same scorecards, comparing ATLAS to GenCast. When trained with the stochastic interpolant variant, ATLAS consistently outperforms GenCast on both CRPS and ensemble mean RMSE out to lead times of approximately seven days, the time scales where the predictability of the atmosphere is most actionable. On longer time scales the skill differences between ATLAS-SI and GenCast are statistically indistinguishable at our chosen sampling. The two other variants of ATLAS also show interesting skill with subtle channel- and lead-time-specific tradeoffs. ATLAS-EDM has virtually equivalent skill as GenCast for all variables except the 500 hPa geopotential, for which it underperforms. ATLAS-CRPS has detectable skill gains clustered around 8-10 day lead times, despite underperforming during the first two days.

15-day rollout comparison

For a complementary view, Figure 5 shows time series comparing the probabilistic error growth of ATLAS (solid lines) to the IFS (dotted) and GenCast (dashed) ensemble baselines, as measured by ensemble mean RMSE (left) and CRPS (middle) while also revealing the spread-error ratio (right) of the underlying ensembles.

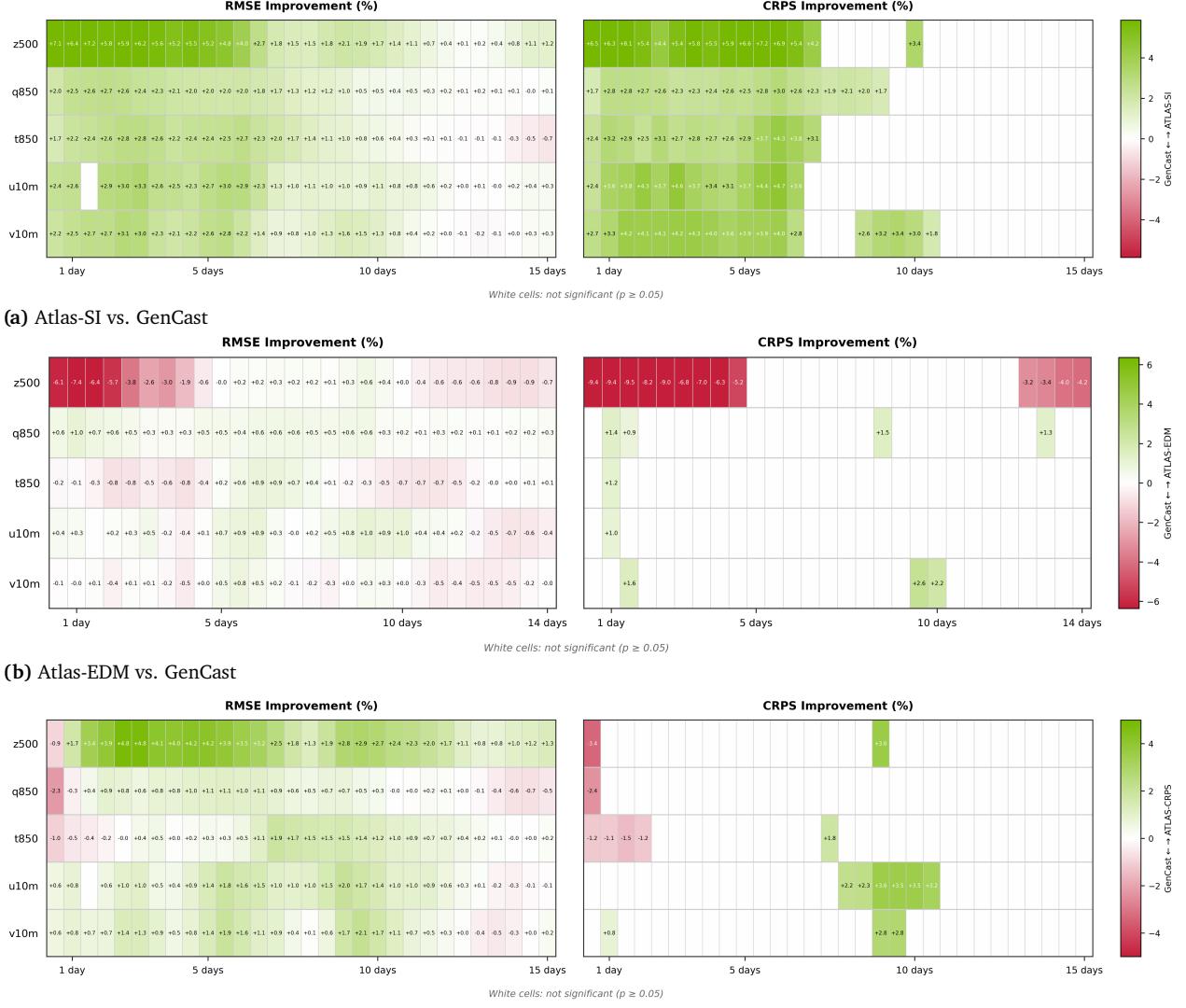


Figure 4: Comparing Atlas and GenCast across RMSE and CRPS with statistical significance ($p < 0.05$), for all three variants, SI, EDM and CRPS. Green means Atlas is better, red means GenCast is better, white means the difference is not statistically significant.

Consistent with the skill difference scorecards, ATLAS-SI tends to exhibit lower ensemble RMSE and CRPS for most variables during the initial forecast week, despite being slightly more under-dispersive than the GenCast baseline, which is often over-dispersive at longer lead times.

We observe in Figure 5 that SSR of ATLAS -CRPS remains under-dispersive compared to the other approaches, a behavior consistent with results in FourCastNet 3 [14]. The CRPS objective can be interpreted as a min-max game: minimizing the error term encourages accuracy, while minimizing the negative interaction term encourages ensemble diversity. While this specific balance of attractive and repulsive forces is necessary for the CRPS to be a proper scoring rule, these results suggest that this competition is uneven. The optimization dynamics appear dominated by the minimization of the first-moment error, implying that the repulsive force intended to drive ensemble spread provides a comparatively weaker signal. Consequently, the model converges to a state that prioritizes minimizing the forecast error over maximizing the ensemble dispersion, leaving the ensemble under-dispersed despite the theoretical incentive for equilibrium.

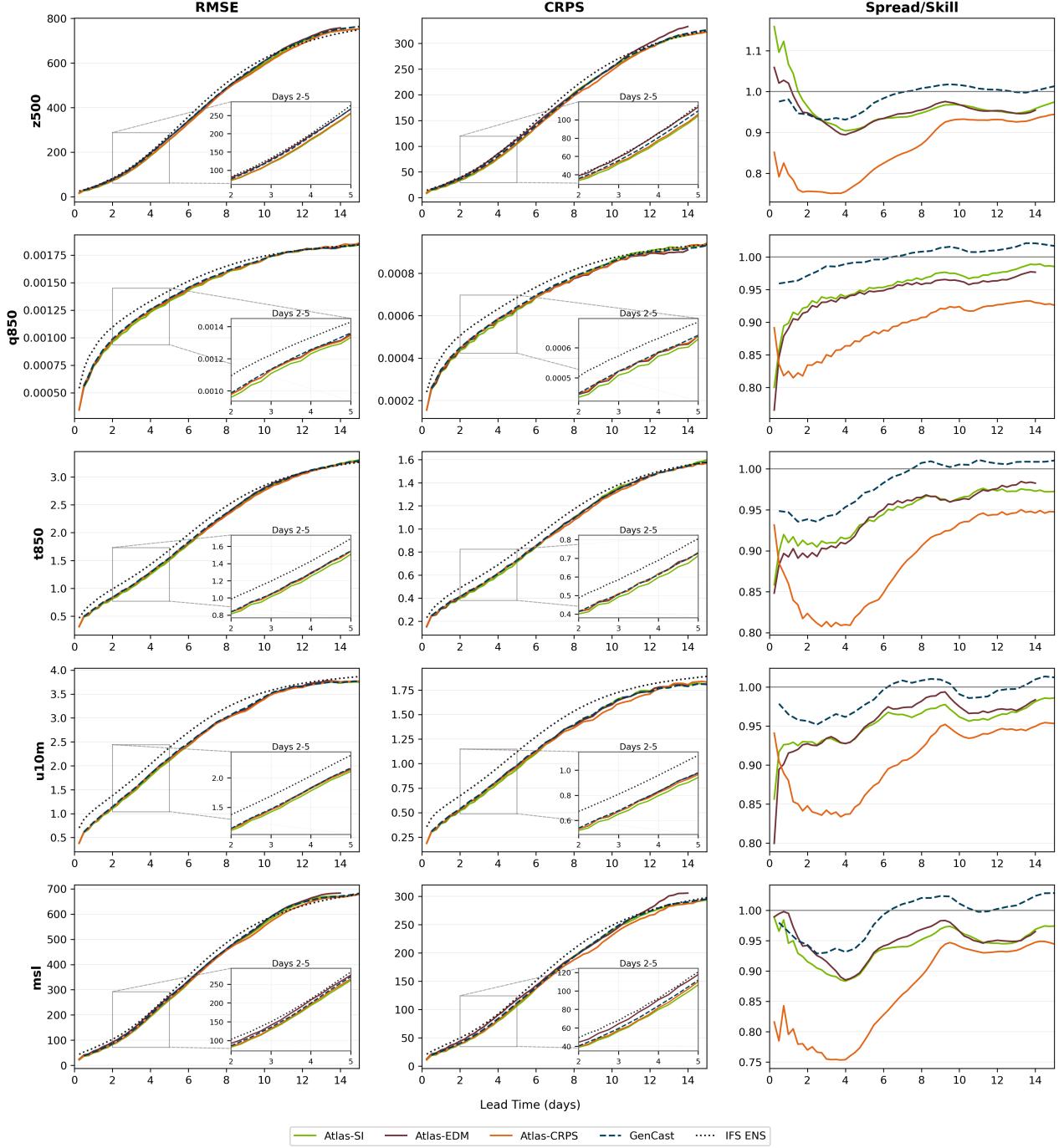


Figure 5: 15-day rollout comparison between ATLAS, IFS, and GenCast.

4.2. Case Study: Storm Dennis

Case study analysis is important to complement headline score statistics that can otherwise obscure issues of realistic spatial coherence in atmospheric model validation.

Here we examine the performance of ATLAS-SI on one of most intense extratropical cyclones ever recorded, Storm Dennis, which underwent explosive cyclogenesis on 13 February 2020, reached its minimum central pressure on the 14th and produced destructive winds and heavy precipitation over the British Isles on the 15–16th. Ensemble forecasts were initialized at 00 UTC on 11 February 2020. Figure 6 shows 500-hPa geopotential height and 850-hPa wind speed at forecast lead times of 2–5 days for two ensemble members,

along with the ERA5 analysis. The color scale for wind speed transitions to orange around 33 m s^{-1} , which is the threshold for 1-minute-sustained surface winds in a Category 1 hurricane. Both ensemble members perform well, capturing Dennis' rapid intensification and the full strength of the evolving wind field at lead times from 2-4 days. The 5-day forecast is also strong, with the caveat that the wind speed is just slightly under-forecast over Wales and westward across the Atlantic.

Also shown in the bottom row of Figure 6 are normalized spectra of the 850-hPa horizontal kinetic energy plotted as a function of total spherical wavenumber. At all lead times, the spectra for both ensemble members are in excellent agreement with that for ERA5. Despite some modest systematic under-representation of variance at the finest length scales (horizontal wavenumbers greater than 200) there is no lead-time dependent trend in the evolution of the spectral bias, which could be viewed as consistent with maintenance of encouragingly self-similar dynamical phenomena throughout the forecast.

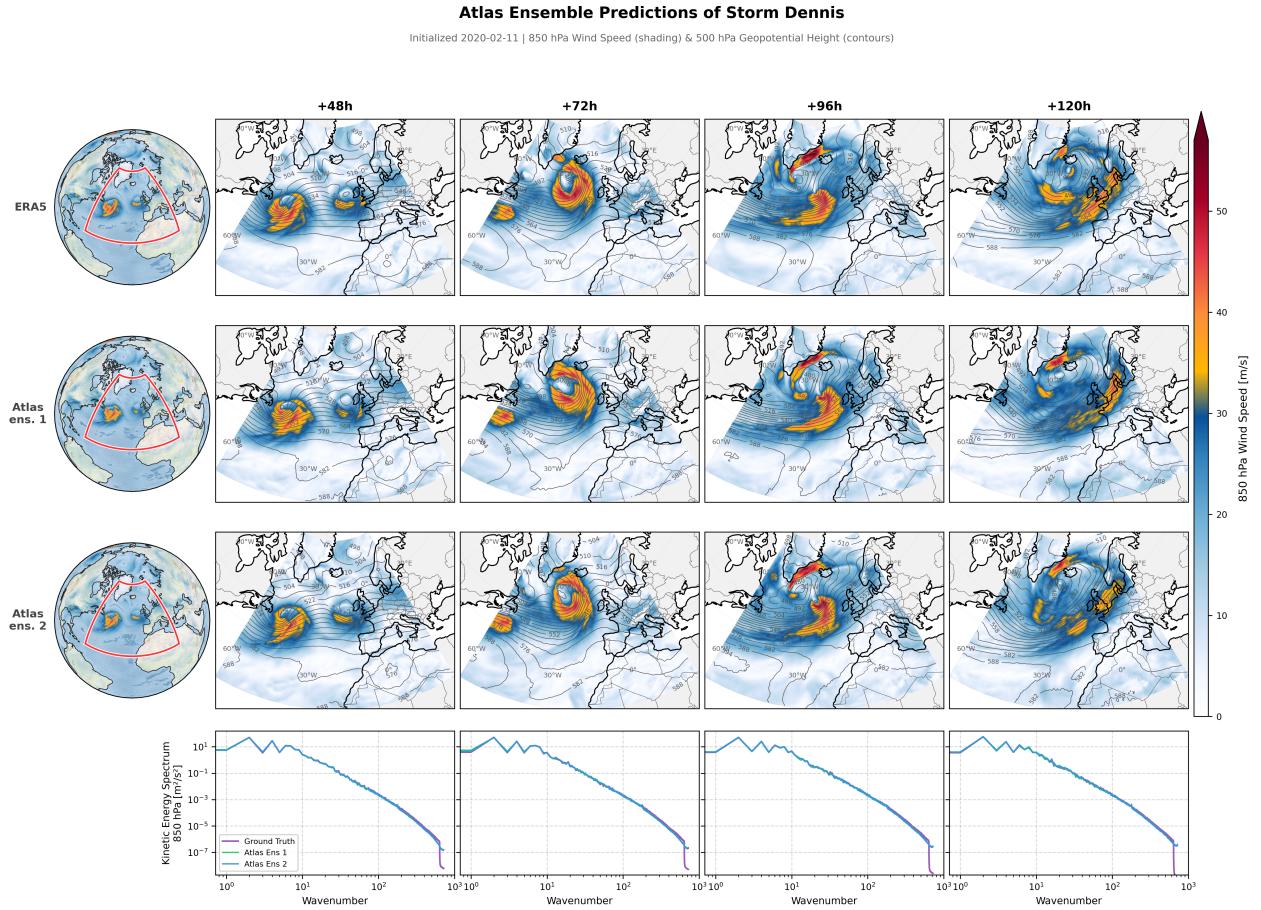


Figure 6: Atlas-SI forecast for storm Dennis. 850-hPa wind speed and 500-hPa geopotential height (dam) for the ERA5 ground truth, predictions from ensemble members one and two, and angular power spectral densities from top to bottom.

4.3. Tropical cyclone evaluation

We also evaluate ATLAS performance on tropical cyclone (TC) tracking, focusing on just the most skillful ATLAS-SI variant, and comparing the model to GenCast, a diffusion model with strong TC tracking performance. We use the 2020 subset of International Best Track Archive for Climate Stewardship (IBTRACs) dataset [54] for ground-truth track data. Of the 118 storms that year, we filter the dataset in two ways for a fair intercomparison. First, to ensure examples are independent, we only use one initialization time per storm, which is the first time IBTRACs reports on the storm. Second, to evaluate on an equal number of samples for each lead time out to seven days, we include only tropical cyclone tracks whose lifetimes extended to at least seven days. The 46

storms that satisfy these criteria form the test set. For the metrics presented, we evaluate each model on 32 ensemble members.

As tracking algorithms often have a substantial effect on overall tracking performance, for fair comparison we use a standard tracker, TempestExtremes [55]. Since ATLAS-SI matches the assumptions of the TempestExtremes tracker by using a 6-hour timestep, we use the standard hyperparameters. As GenCast uses a 12-hour timestep, however, we follow the protocol in the original GenCast paper [11] and use those hyperparameters to generate tracks.

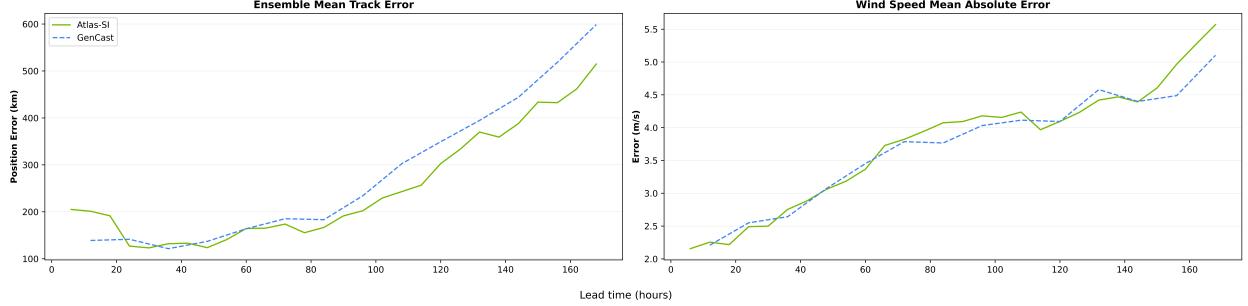


Figure 7: Tropical Cyclone Tracking: Comparison of ATLAS-SI and GenCast on all 46 tropical storms with at least seven day lifetimes during the held out year of 2020. ATLAS-SI and GenCast are represented in green and dashed blue, respectively. **Left:** Ensemble Mean Track Error. **Right:** Mean Absolute Error of wind speed.

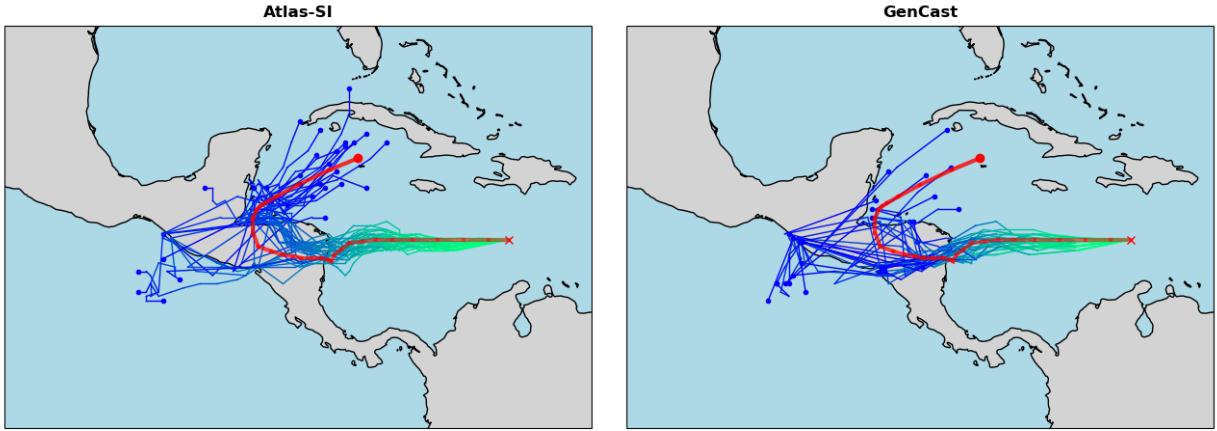


Figure 8: Tropical Cyclone Tracking: Comparison of spaghetti plots of ATLAS-SI and GenCast for the first seven days of Hurricane Eta. Models are initialized at October 31, 2020 UTC 18:00. Origination and termination of the ground truth tropical cyclone track (red) are represented by a “x” and “.” respectively. For predicted tracks, origination is represented by neon green, while termination is represented by blue. **Left:** ATLAS-SI. **Right:** GenCast.

As shown in Figure 7, both the ensemble mean track error and the average intensity error of ATLAS-SI are similar to that of GenCast. ATLAS-SI’s ensemble mean track error improves on GenCast beyond 80 hours and is similar at shorter lead times. The average intensity error of ATLAS-SI matches GenCast’s to 144 hours but slightly lags beyond that point.

Furthermore, ATLAS-SI can generate TC track predictions qualitatively different to, and in many cases more accurate than, GenCast. Figure 8, which shows seven-day “spaghetti” plots of Hurricane Eta for ATLAS-SI and GenCast, is one such example. Hurricane Eta was a Category 4 tropical cyclone from late October to early November 2020, and caused at least 175 fatalities and an estimated \$7.2 billion of damage [56]. Cyclogenesis occurred on October 31 and Eta intensified to a Category 4 on November 3. Here, we show the first seven days of prediction, before Eta made landfall in Cuba and the United States. Of particular interest is the eastward swing after the hurricane makes landfall in Nicaragua, as this would presage likely candidates for subsequent

landfalls. Many more of the ATLAS-SI tracks accurately predict this eastward swing compared to GenCast. Overall, ATLAS-SI predicts that Hurricane Eta was more likely to swing eastward rather than travel westward, while the GenCast predictions place a higher probability of a westward trajectory.

A second example is Tropical Cyclone Damien, which was a Category 2 storm formed on February 3, 2020 and dissipated on February 9, and was the strongest cyclone to make landfall in Western Australia since 2013 [57]. As shown in Figure 9, GenCast predictions exhibit a significant eastward bias, with the model predicting landfall east of observed track. By contrast, ATLAS-SI's predictions show a less eastward bias, with a number of ensemble members mirroring the observed track.

Finally, for completeness, we also include a weaker storm. Tropical Storm Krovanh (also known as Tropical Depression Vicky) originated on Dec 17, 2020 and terminated on December 25 of that year, and caused 9 fatalities and \$4.5M of damage [58]. Figure 10 shows the predicted tracks for the two models; ATLAS-SI's span the full length of the tropical cyclone, while GenCast's terminate before reaching the southern shores of Vietnam and Cambodia.

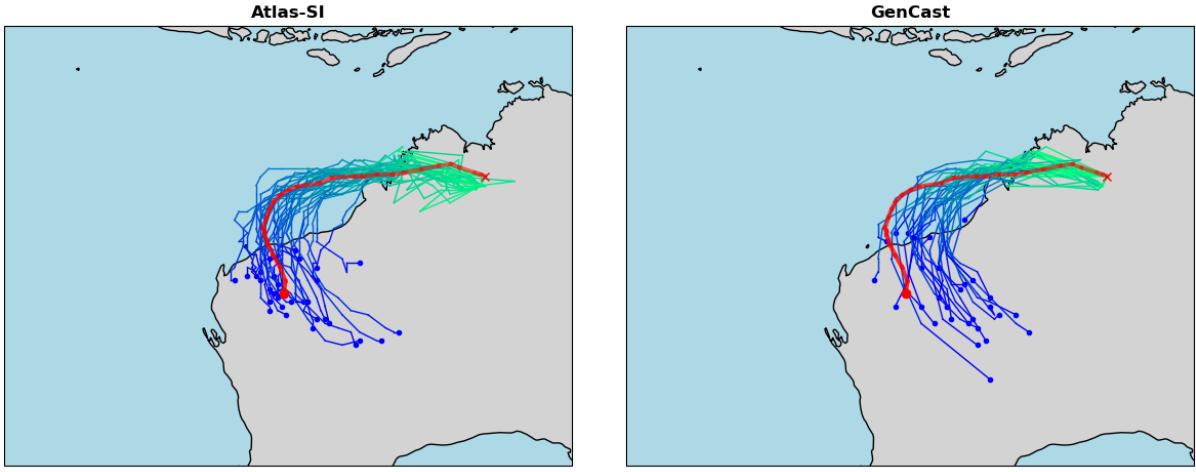


Figure 9: Cyclone Damien Tracking: Comparison of spaghetti plots of ATLAS-SI and GenCast for the first seven days of Cyclone Damien. Models are initialized at February 3, 2020 UTC 00:00. Origination and termination of the ground truth tropical cyclone track (red) are represented by a “x” and “.” respectively. For predicted tracks, origination is represented by neon green, while termination by blue. **Left:** ATLAS-SI. **Right:** GenCast.

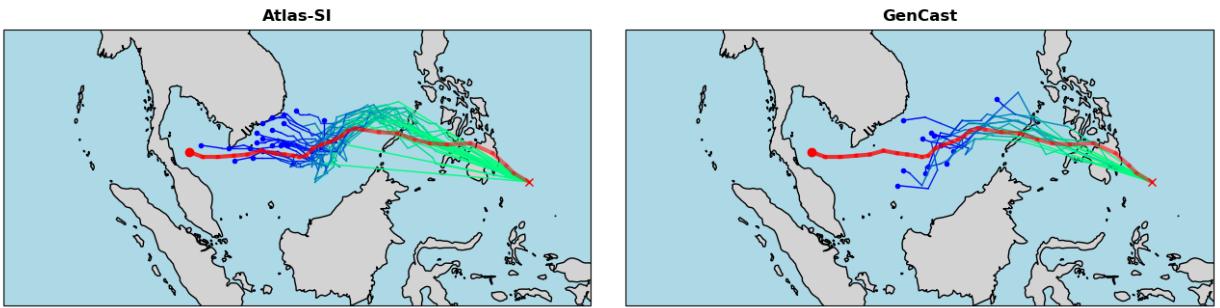


Figure 10: Tropical Storm Krovanh Tracking: Comparison of spaghetti plots of ATLAS-SI and GenCast for the first seven days of Tropical Storm Krovanh. Models are initialized at December 17, 2020 UTC 12:00. Origination and termination of the ground truth tropical cyclone track (red) are represented by a “x” and “.” respectively. For predicted tracks, origination is represented by neon green, while termination by blue. **Left:** ATLAS-SI. **Right:** GenCast.

5. Validating latent space modeling

While we have advocated for simplifying and using standard architectures for AI weather models, one area where we found a more bespoke approach beneficial is the design of the encoder and decoder. Here, we provide empirical evidence justifying our choices.

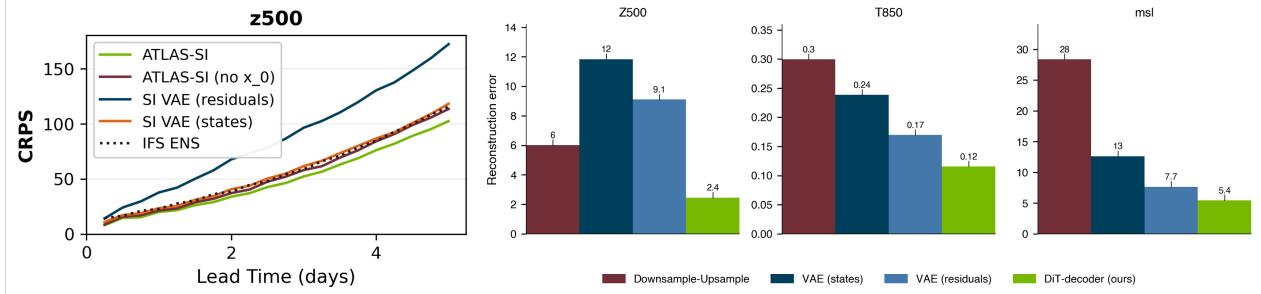


Figure 11: **Comparison of learned latent and downsampled latent.** While 6– and 12–hour performance of forecasts learned in a learned latent space are competitive, errors in auto-regressive forecasts accumulate at a faster rate.

The predominant approach to latent modeling in image and video domains is to compress the data with a variational autoencoder. The variables that constitute our atmospheric state vector, however, exhibit quite different properties from the RGB channels that constitute images and video. In particular, variables such as temperature, wind, and geopotential are far more heterogeneous than RGB data. As a result, as mentioned in subsection 2.2, standard VAE modeling leads to less well-behaved latent spaces. Moreover, since weather prediction inherently is a “pixel” prediction task, a decoder with low reconstruction error is an important desideratum. Here, we empirically demonstrate that even a well-tuned VAE with low reconstruction error leads to less accurate forecasts.

Baseline autoencoding models

The learned autoencoder was tuned to achieve the lowest reconstruction error of all 16×-spatially compressed latent models. The structure of the model is as follows: after applying sinusoidal positional encodings to mean-variance normalized inputs, a convolution layer embeds these inputs into spatial resolution of (195×390) , but increases the hidden dimension to 832. This embedded state is then processed by eight DiT blocks, with each DiT block’s linear layer progressively decreasing the hidden dimension by 112 channels. The dimensionality of the latents after the eighth block is $64 \times 195 \times 390$. The decoder mirrors the encoder with eight DiT blocks and a final linear layer. Each DiT block’s linear layer progressively increases the hidden dimension, while the final linear layer projects the data back to the original dimensionality.

The $64 \times 195 \times 390$ state is the latent used for predictive modeling. We train stochastic interpolants models, and evaluate two different prediction approaches. The first predicts the next latent state given the current latent one, while the second the next latent residual given the current latent state. A second autoencoder, which takes as input and output state residuals, is trained to model the latent residual.

Figure 11 shows that, even though reconstruction errors for the state-to-state and residual-to-residual models are relatively low for T850 and Q850 — Z500 error is relatively high — and while 6- and 12-hour prediction accuracy was competitive, longer rollout performance lagged IFS ENS. Furthermore, and perhaps somewhat surprisingly, even though residual-to-residual reconstruction error is lower than for state-to-state, the resulting autoregressive rollout accumulated error at a higher rate compared to the latent state prediction model.

Streamlining the encoding and decoding process

We found that a trivial “encoding” of bilinear interpolation preserves the structure of the latent space at the cost of small-scale information. As shown in Figure 11, however, extending this approach by trivially “decoding” the compressed state using bilinear upsampling yields very high reconstruction errors. Instead, we train a

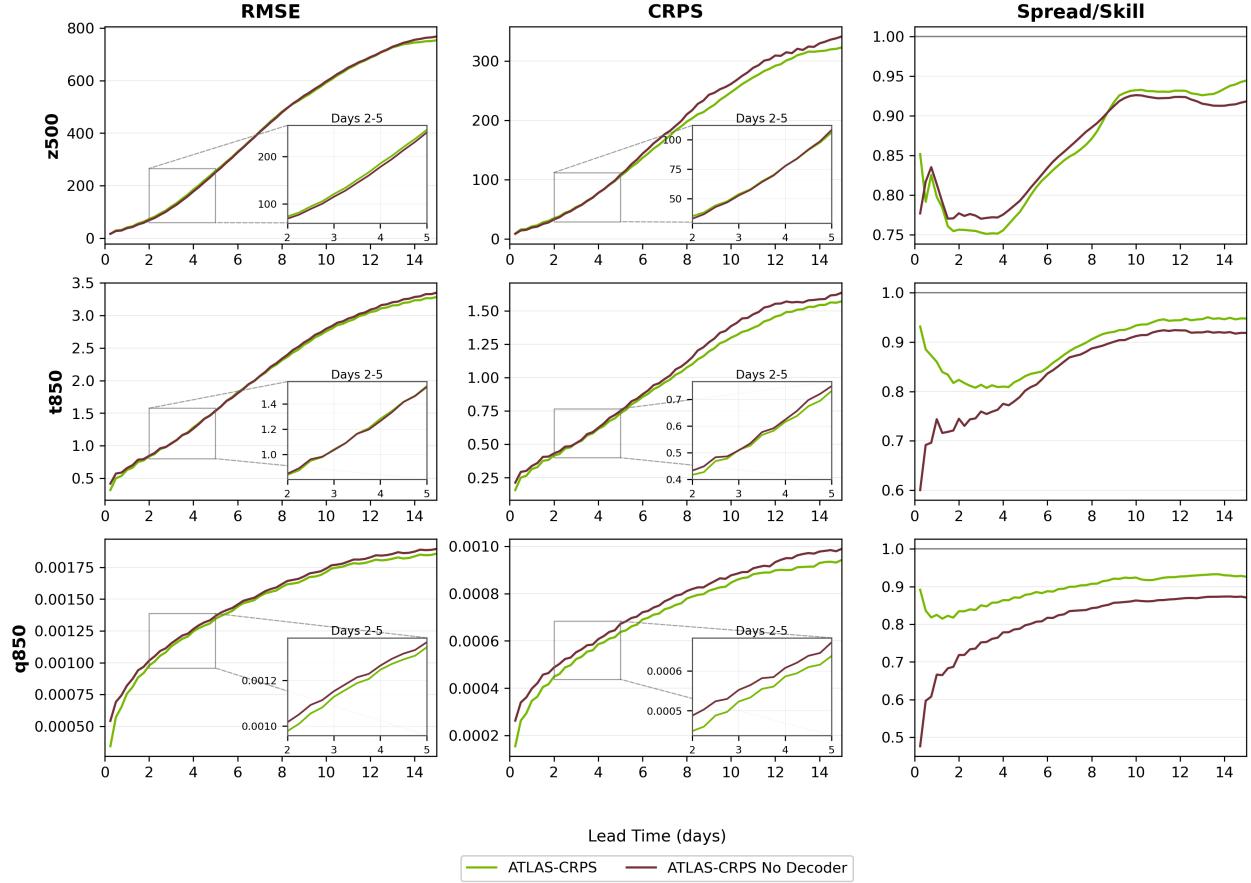


Figure 12: **Importance of the decoder in ATLAS.**

DiT decoder that takes as input the low-resolution residual and the high-resolution input state and predicts the high-resolution residual. One can think of the decoder as predicting a high-resolution residual consistent with low-resolution residual and high-resolution input. This method not only dramatically decreases the reconstruction error, achieving the lowest error among all models, but also yields the lowest *rollout* error among all tested methods.

We note that, even though the DiT backbone only ever operates on low-resolution states, the decoder at each time step retains high-resolution information of the previous timestep. As a result the model retains high-resolution temporal consistency during inference.

6. Conclusion

In this paper, we challenge the prevailing view that probabilistic weather forecasting requires complex, domain-specific architectures or training methods. We proposed ATLAS, a simple, scalable framework—comprising of a latent transformer and a local projector. Combined with any probabilistic estimation framework, from stochastic interpolants to diffusion models and CRPS ensemble training, we show that it outperforms both the operational IFS and the deep learning model GenCast.

We speculate that two key ingredients in these successes have been scaling and an explicitly multi-scale signal processing framework. During initial development, we found that progressively increasing the computational complexity of the backbone model consistently improved forecasting skill, seemingly promoting the model’s ability to learn physical correlations naturally. From this view, the intrinsic scalability of transformer architectures and ample tooling from their scaling in non-weather domains might portend a future of convergence towards ATLAS-like architectures. Furthermore, our approach decouples the tasks of learning the time evolution of the full-resolution state from the prediction of the most fundamental atmospheric dynamics that

take place at larger horizontal scales, allowing us to separately scale the models for each task.

We conduct an extensive evaluation of our models and demonstrate statistically significant improvements in aggregate metrics, sharp emulation of extreme events, and accurate tracking of tropical cyclones all at a reduced computational cost compared to GenCast. These results suggest that the path to higher fidelity lies not in architectural complexity, but in the scaling of general-purpose foundation models.

We readily admit several limitations of our work. Our baseline – while intentionally chosen for a combination of known strength and feasibility of intercomparison under the constraint of fully open ERA5 training data – is not the only interesting strong baseline that would be worth measuring skill against. Fine-tunings of ATLAS beyond the corpus of ERA5 data on operational IFS analyses would be useful towards eventual comparisons against more contemporary baselines of significance such as AIFS and FGN [12, 13], which might prove revealing. Likewise, additional validation of phenomena beyond tropical and midlatitude cyclones, and analysis beyond the held-out year of 2020, is recommended towards greater understanding of the behavior of ATLAS and for measuring its statistical performance on extremes.

Meanwhile, one contribution of the work we suspect may have general implications is the progress in achieving frontier skill using diffusion-based approaches, which has historically required some compromise. For instance, there seems to be a trade-off between short and medium-range skill. While diffusion models can be trained with wide noise schedules to capture tail behavior and achieve excellent short-term forecasts (e.g., one week), this often degrades skill at longer lead times. Thus, we deliberately sacrificed some near-term skill to improve stability. We believe this can be mitigated through improved noise-schedule design and tighter overfitting control.

Finally, since diffusion is readily controllable, successful weather prediction built on it could be viewed as opening up paths towards guided forecasting, such as to a subset of the ensemble that validates as a particular regional hazard of outsize interest, as demonstrated in [26]. Therefore this technical progress may prove helpful to broader efforts across the community to evolve AI weather models from narrow, task-specific technologies to broader, task-diverse foundational informatics systems.

References

- [1] 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. *arXiv preprint arXiv:2202.11214*, 2 2022. [1](#)
- [2] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*, 2023. [1](#)
- [3] Ryan Keisler. Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*, 2022. [1](#)
- [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*, 619:533–538, 7 2023. [1](#)
- [5] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Alexander Pritzel, Suman Ravuri, Timo Ewalds, Ferran Alet, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Jacklynn Stott, Oriol Vinyals, Shakir Mohamed, and Peter Battaglia. Graphcast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 12 2022. [1](#)
- [6] Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. *Proceedings of the 40th International Conference on Machine Learning*, 202:2806–2823, 6 2023. [1](#), [7](#)
- [7] Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, et al. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*, 2023. [1](#)

- [8] 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. *arXiv preprint arXiv:2311.07222*, 11 2023. [1](#)
- [9] Ankur Mahesh, William Collins, Boris Bonev, Noah Brenowitz, Yair Cohen, Joshua Elms, Peter Harrington, Karthik Kashinath, Thorsten Kurth, Joshua North, Travis OBrien, Michael Pritchard, David Pruitt, Mark Risser, Shashank Subramanian, and Jared Willard. Huge ensembles part i: Design of ensemble weather forecasts using spherical fourier neural operators. *arXiv preprint arXiv:2408.03100*, 8 2024. [1](#)
- [10] Ankur Mahesh, William Collins, Boris Bonev, Noah Brenowitz, Yair Cohen, Peter Harrington, Karthik Kashinath, Thorsten Kurth, Joshua North, Travis OBrien, Michael Pritchard, David Pruitt, Mark Risser, Shashank Subramanian, and Jared Willard. Huge ensembles part ii: Properties of a huge ensemble of hindcasts generated with spherical fourier neural operators. *arXiv preprint arXiv:2408.01581*, 8 2024. [1](#)
- [11] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025. [1](#), [2](#), [16](#)
- [12] Simon Lang, Mihai Alexe, Mariana C. A. Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Boual-lègue, 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. *arXiv preprint arXiv:2412.15832*, 12 2024. [1](#), [2](#), [6](#), [7](#), [8](#), [20](#)
- [13] Ferran Alet, Ilan Price, Andrew El-Kadi, Dominic Masters, Stratis Markou, Tom R Andersson, Jacklynn Stott, Remi Lam, Matthew Willson, Alvaro Sanchez-Gonzalez, et al. Skillful joint probabilistic weather forecasting from marginals. *arXiv preprint arXiv:2506.10772*, 2025. [1](#), [2](#), [7](#), [8](#), [20](#)
- [14] Boris Bonev, Thorsten Kurth, Ankur Mahesh, Mauro Bisson, Jean Kossaifi, Karthik Kashinath, Anima Anandkumar, William D Collins, Michael S Pritchard, and Alexander Keller. Fourcastnet 3: A geometric approach to probabilistic machine-learning weather forecasting at scale. *arXiv preprint arXiv:2507.12144*, 2025. [1](#), [2](#), [13](#)
- [15] Xiaohui Zhong, Lei Chen, Hao Li, Roberto Buizza, Jun Liu, Jie Feng, Zijian Zhu, Xu Fan, Kan Dai, Jing jia Luo, Jie Wu, and Bo Lu. Fuxi-ens: A machine learning model for efficient and accurate ensemble weather prediction. *Science Advances*, 11(44):eadu2854, 2025. [1](#)
- [16] Oliver Watt-Meyer, Gideon Dresdner, Jeremy McGibbon, Spencer K. Clark, Brian Henn, James Duncan, Noah D. Brenowitz, Karthik Kashinath, Michael S. Pritchard, Boris Bonev, Matthew E. Peters, and Christopher S. Bretherton. Ace: A fast, skillful learned global atmospheric model for climate prediction. *arXiv preprint arXiv:2310.02074*, 10 2023. [1](#)
- [17] Salva Ruhling Cachay, Brian Henn, Oliver Watt-Meyer, Christopher S Bretherton, and Rose Yu. Probabilistic emulation of a global climate model with spherical dyffusion. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 127610–127644, 2024. [1](#)
- [18] Oliver Watt-Meyer, Brian Henn, Jeremy McGibbon, Spencer K Clark, Anna Kwa, W Andre Perkins, Elynn Wu, Lucas Harris, and Christopher S Bretherton. Ace2: accurately learning subseasonal to decadal atmospheric variability and forced responses. *npj Climate and Atmospheric Science*, 8(1):205, 2025. [1](#)
- [19] Haiwen Guan, Troy Arcomano, Ashesh Chattopadhyay, and Romit Maulik. Lucie: A lightweight uncoupled climate emulator with long-term stability and physical consistency for o(1000)-member ensembles. *arXiv preprint arXiv:2405.16297*, 4 2025. [1](#)
- [20] Nathaniel Cresswell-Clay, Bowen Liu, Dale Durran, Zihui Liu, Zachary I. Espinosa, Raul Moreno, and Matthias Karlbauer. A deep learning earth system model for efficient simulation of the observed climate. *arXiv preprint arXiv:2409.16247*, 2 2025. [1](#)

- [21] Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcmano, Romit Maulik, Veerabhadra Kotamarthi, Ian Foster, Sandeep Madireddy, and Aditya Grover. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. *arXiv preprint arXiv:2312.03876*, 2024. 1
- [22] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1
- [23] Ira J. S. Shokar, Rich R. Kerswell, and Peter H. Haynes. Stochastic latent transformer: Efficient modeling of stochastically forced zonal jets. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004177, 2024. e2023MS004177 2023MS004177. 2, 7
- [24] Guillaume Couairon, Renu Singh, Anastase Charantonis, Christian Lessig, and Claire Monteleoni. Archeweather & archesweathergen: a deterministic and generative model for efficient ml weather forecasting. *arXiv preprint arXiv:2412.12971*, 12 2024. 2
- [25] Jason Stock, Troy Arcmano, and Rao Kotamarthi. Swift: An autoregressive consistency model for efficient weather forecasting, 2025. 2
- [26] Noah D Brenowitz, Tao Ge, Akshay Subramaniam, Peter Manshausen, Aayush Gupta, David M Hall, Morteza Mardani, Arash Vahdat, Karthik Kashinath, and Michael S Pritchard. Climate in a bottle: Towards a generative foundation model for the kilometer-scale global atmosphere. *arXiv preprint arXiv:2505.06474*, 2025. 2, 20
- [27] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021. 4, 5
- [28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 4, 5
- [29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4
- [30] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 2019. 4
- [31] Yifan Zhou, Zeqi Xiao, Shuai Yang, and Xingang Pan. Alias-free latent diffusion models: Improving fractional shift equivariance of diffusion latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 4
- [32] Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. 4
- [33] Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025. 4
- [34] Madalina Surcel, Isztar Zawadzki, and MK Yau. A study on the scale dependence of the predictability of precipitation patterns. *Journal of the Atmospheric Sciences*, 72(1):216–235, 2015. 4
- [35] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 5
- [36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

- [37] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 5
- [38] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5
- [39] A. Gettelman, D. J. Gagne, C.-C. Chen, M. W. Christensen, Z. J. Lebo, H. Morrison, and G. Gantos. Machine learning the warm rain process. *Journal of Advances in Modeling Earth Systems*, 13(2), 2021. 5
- [40] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 6
- [41] Yifan Chen, Mark Goldstein, Mengjian Hua, Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Probabilistic forecasting with stochastic interpolants and follmer processes. *arXiv preprint arXiv:2403.13724*, 2024. 6
- [42] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 6, 7
- [43] AJ Roberts. Modify the improved euler scheme to integrate stochastic differential equations. *arXiv preprint arXiv:1210.0933*, 2012. 6
- [44] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 6, 7
- [45] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021. 7
- [46] Thomas A. Brown. Admissible scoring systems for continuous distributions. *RAND Corporation, Santa Monica, CA*, 1974. 7
- [47] James E. Matheson and Robert L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, 1976. 7
- [48] Tilman Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. 7
- [49] Eviatar Bach, Ricardo Baptista, Daniel Sanz-Alonso, and Andrew Stuart. Inverse problems and data assimilation: A machine learning approach. *arXiv preprint arXiv:2410.10523*, 2024. 7
- [50] Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd. *Advances in neural information processing systems*, 28, 2015. 8
- [51] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 8
- [52] Michaël Zamo and Philippe Naveau. Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. *Math. Geosci.*, 50(2):209–234, February 2018. 10
- [53] V. Fortin, M. Abaza, F. Anctil, and R. Turcotte. Why should ensemble spread match the rmse of the ensemble mean? *Journal of Hydrometeorology*, 15(4):1708–1713, 2014. 10
- [54] Kenneth R Knapp, Michael C Kruk, David H Levinson, Howard J Diamond, and Charles J Neumann. The international best track archive for climate stewardship (ibtracs) unifying tropical cyclone data. *Bulletin of the American Meteorological Society*, 91(3):363–376, 2010. 15
- [55] Paul A Ullrich, Colin M Zarzycki, Elizabeth E McClenney, Marielle C Pinheiro, Alyssa M Stansfield, and Kevin A Reed. Tempestextremes v2. 1: A community framework for feature detection, tracking, and analysis in large datasets. *Geoscientific Model Development*, 14(8):5023–5048, 2021. 16

- [56] Wikipedia contributors. Hurricane eta. https://en.wikipedia.org/wiki/Hurricane_Eta, 2020. Accessed: 2025-01-20. 16
- [57] Wikipedia contributors. Severe tropical cyclone damien. https://en.wikipedia.org/wiki/Cyclone_Damien, 2020. Accessed: 2025-01-20. 17
- [58] Wikipedia contributors. Tropical storm krovanh. [https://en.wikipedia.org/wiki/Tropical_Storm_Krovanh_\(2020\)](https://en.wikipedia.org/wiki/Tropical_Storm_Krovanh_(2020)), 2020. Accessed: 2025-01-20. 17