Otter Weather: Skillful and Computationally Efficient Medium-Range Weather Forecasting

Cristiana Diaconu*
University of Cambridge
cdd43@cam.ac.uk

Jonas Scholz*
University of Cambridge
js2731@cam.ac.uk

Aliaksandra Shysheya
University of Cambridge

Stratis Markou
University of Cambridge

Payel Mukhopadhyay
University of Cambridge

Miles Cranmer
University of Cambridge

Richard E. Turner
University of Cambridge

Abstract

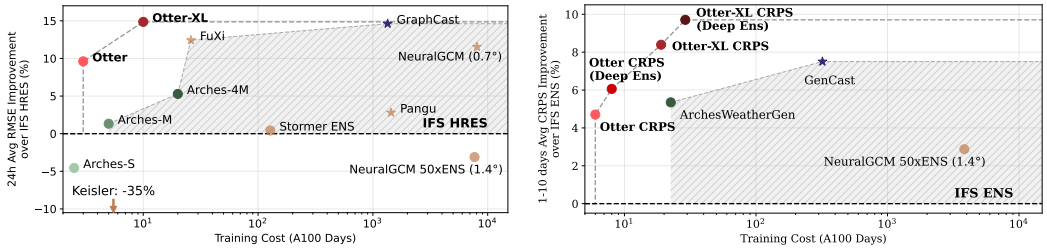
State-of-the-art medium-range AI weather models can outperform traditional Numerical Weather Prediction (NWP) but require massive training budgets. This restricts usage for under-resourced groups and severely limits fast model iteration. Here we develop Otter Weather, a highly efficient spatiotemporal forecasting model designed to make high-performance AI weather prediction accessible beyond well-resourced laboratories. Evaluated on ERA5 reanalysis data at 1.5° resolution using standard WeatherBench protocols, the Otter family significantly advances the skill-compute Pareto frontier. The deterministic version outperforms the best NWP baseline by 9.6% at a 24-hour lead time while requiring fewer than 3.5 A100-days for training. It provides a 2× efficiency gain over lightweight AI models and a 100-fold reduction in compute compared to resource-intensive frontier architectures. We extend these efficiency gains into probabilistic forecasting by training via the Continuous Ranked Probability Score (CRPS). Scaling to a larger architecture, Otter-XL achieves a 9.7% CRPS improvement over the IFS ENS baseline. This yields an almost two-fold increase in predictive skill over comparable lightweight models at similar compute budgets. Otter-XL also outperforms frontier architectures like GenCast by over 2%, while using an order of magnitude less compute. Finally, we show that Otter scales to 0.25°: post-hoc downscaling of Otter-XL matches GraphCast's skill from 72 h onward at negligible additional cost, and native 0.25° training requires only a change of patch size to reach competitive skill at 17 A100-days.

1 Introduction

The field of global weather forecasting is being transformed by data-driven approaches. AI models have reached performance levels comparable to strong numerical weather prediction (NWP) baselines, even exceeding them in many tasks [Lam et al., 2023, Lang et al., 2024, Alet et al., 2025, Bodnar et al., 2025]. This paradigm shift holds the potential to democratise weather forecasting, but fully realising this vision now rests on reducing training costs so that groups outside of well-resourced AI laboratories can design, train and control their own models.

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Deterministic Models

(b) Probabilistic Models

Figure 1: **Skill score over IFS HRES/ENS on the headline variables.** (a) RMSE at 24h. (b) CRPS averaged over 1-10 days. Circles (○) represent models trained at lower resolutions (1°/1.4°/1.5°), while stars (⋆) indicate higher-resolution training (0.25°/0.7°). The models in the **Otter Weather** family advance both the deterministic and probabilistic Pareto frontiers of skill versus compute.

Lowering the training cost barrier would allow a diverse range of beneficiaries—from academic institutions and under-resourced operational agencies to startups—to advance the field in two complementary ways. First, it enables such groups to train competitive weather models from scratch rather than being restricted to fine-tuning released foundation models. This pre-training capability is essential for scientific independence: it lets researchers rigorously audit, reproduce, and extend models without relying on the original developers, and to tailor architectures to specific regions, variables, or downstream tasks. Second, regardless of available resources, a rapid iteration cycle accelerates model development, retraining and deployment for all practitioners, and speeds up scientific discovery.

At present, however, the field remains far from this level of accessibility. While the community is becoming less reliant on the CPU-based supercomputers required for traditional NWP, current state-of-the-art (SOTA) data-driven methods [Lam et al., 2023, Price et al., 2025, Alet et al., 2025] have established a new barrier to entry: they require massive, distributed clusters of hundreds of GPUs or TPUs—infrastructure accessible only to well-resourced AI laboratories. In this work, we investigate whether this paradigm can be shifted to make high-performance weather forecasting accessible to practitioners with limited resources, down to a single GPU. To do so, we first identify the primary drivers of current computational costs. Many SOTA architectures rely on heavy inductive biases, such as custom graph neural networks (GNNs) [Lam et al., 2023, Price et al., 2025] or complex spherical geometry operators [Mahesh et al., 2025]. While physically motivated, these specialised components often constrain the model and lack the hardware optimisation of mainstream techniques used in Large Language Models (LLMs) and computer vision.

To date, the most efficient competitive models are ArchesWeather [Couairon et al., 2024], ERDM [Cachay et al., 2025], and U-Cast [Cachay et al., 2026]. While the deterministic ArchesWeather model requires 5 A100-days of training, probabilistic models like ArchesWeatherGen and U-Cast demand over 20, and ERDM exceeds 40. We push this further by building on general-purpose components, with no Earth-specific inductive biases: a standard 2D Swin Transformer [Liu et al., 2021] with Rotary Positional Embeddings (RoPE) [Su et al., 2024] and SwiGLU activations [Shazeer, 2020], trained with the Muon optimiser [Jordan et al., 2024]. These components are proven to scale in other domains and come with mature hardware support and low-level optimisations. With them, Otter advances both the deterministic and probabilistic skill–compute Pareto frontiers in weather modelling.

Our core contributions are:

1. **A Compute-Efficient Deterministic Backbone.** We introduce **Otter**, a 2D Swin-UNet Transformer model that establishes a new skill-compute Pareto frontier (fig. 1a). Trained on a single GPU in under 3.5 A100 days, Otter outperforms leading low-compute AI forecasting models. Furthermore, our scaled model, **Otter-XL**, surpasses FuXi's [Chen et al., 2023] performance at ∼3x lower compute cost and achieves skill at 1.5° resolution comparable to GraphCast [Lam et al., 2023] while requiring ∼130x less training compute.
2. **Low-Compute Probabilistic Forecasting.** We demonstrate that our architectural advances prove equally effective in the probabilistic setting (fig. 1b), allowing us to significantly advance the probabilistic skill-compute Pareto frontier. At the base scale, Otter yields competitive forecasts with a total compute budget under 8 A100-days — less than half the $>$ 20 A100-day budget of

recent efficient models — outperforming ArchesWeatherGen and remaining highly competitive with U-Cast. With a higher compute budget, yet still less than 30 A100-days, Otter-XL delivers an almost two-fold skill increase over comparable lightweight models, and surpasses resource-intensive frontier architectures like GenCast by over 2% while using an order of magnitude less compute.

3. **Scaling to 0.25°.** We show that Otter reaches competitive skill at ERA5's native $0.25°$ resolution by two routes. A per-grid-cell linear downscaler applied to Otter-XL matches GraphCast's skill from 72h onwards at negligible additional training cost, and a natively trained variant, which requires minimal architectural modifications (differing from Otter-XL only in patch size), comes within 3 points of GraphCast in approximately 17 A100-days.
4. **Reevaluating Inductive Biases & A Practical Guide.** We provide a systematic ablation study demonstrating that standard, hardware-optimised components (RoPE, SwiGLU, Muon) are sufficient for high-skill forecasting. We distill these findings into a practical training recipe for practitioners under resource constraints, while also documenting techniques we explored, but did not pursue, as their marginal gains did not justify their implementation overhead.

2 Related Work

2.1 Data-Driven Global Weather Forecasting

Specialised and Foundation Models. Recent breakthroughs in data-driven weather forecasting have seen learned models match or surpass numerical weather prediction (NWP) baselines. However, models achieving state-of-the-art (SOTA) performance generally rely on heavy, domain-specific architectural modifications. For instance, Pangu-Weather [Bi et al., 2023] uses 3D Earth-specific Transformers, while other models employ complex Graph Neural Networks (GNNs) [Keisler, 2022, Lam et al., 2023, Price et al., 2025, Lang et al., 2024, Alet et al., 2025] or spectral approaches like Spherical Fourier Neural Operators [Pathak et al., 2022, Bonev et al., 2025]. These physical priors often incur massive training overheads—hundreds of GPU-days—and require custom-written kernels [Fu et al., 2024]. Similarly, adapting foundation models such as ClimaX [Nguyen et al., 2023] and Aurora [Bodnar et al., 2025] requires substantial engineering effort and hardware beyond the reach of many groups [Subich, 2025, Lehmann et al., 2025]. In addition to that, building on a pre-trained model implies inheriting its architecture, variable set, and biases. Even when fine-tuning is affordable, this limits how far a model can be tailored to a particular research objective or region.

Efficient Architectures and Probabilistic Modelling. In response, a counter-trend has emerged focusing on compute-efficient architectures that minimise inductive biases, such as Stormer [Nguyen et al., 2024], ArchesWeather [Couairon et al., 2024], and U-Cast [Cachay et al., 2026]. Simultaneously, the field is shifting toward probabilistic modelling to capture predictive uncertainty. While flow-based and diffusion models (e.g., GenCast [Price et al., 2025], ArchesWeatherGen [Couairon et al., 2024]) achieve this, they require computationally expensive iterative denoising at inference time. Alternatively, optimising proper scoring rules like the Continuous Ranked Probability Score (CRPS)—a technique pioneered by FGN [Alet et al., 2025] and AIFS-CRPS [Lang et al., 2026], and inspired by PDE literature [Diaconu et al., 2026]—offers a highly efficient route to fine-tune a deterministic checkpoint into a probabilistic model. U-Cast [Cachay et al., 2026] recently applied this concept, using Monte Carlo (MC) dropout in a U-Net architecture to build a deep ensemble from multiple CRPS fine-tuned checkpoints. However, models like U-Cast and ArchesWeatherGen still operate at training budgets exceeding 20 A100-days. In contrast, our work pushes the Pareto frontier even further by leveraging CRPS training with MC dropout within a strictly lower-compute, transformer-based setting. Rather than undertaking independent fine-tuning runs to construct an ensemble, we use an efficient methodology that reuses checkpoints already generated during standard hyperparameter tuning, achieving competitive probabilistic performance at a fraction of the computational cost.

2.2 ML Breakthroughs from Vision and Language Modelling

As weather models converge with general-purpose architectures, the broader transferability of modern techniques from the LLM and computer vision communities remains under-explored. While ArchesWeather successfully integrated SwiGLU activations [Shazeer, 2020], we argue the field still lacks a systematic evaluation of domain-agnostic adaptations. We ablate Rotary Positional

Embeddings (RoPE) [Su et al., 2024] and the Muon optimiser [Jordan et al., 2024], and briefly report on preliminary experiments with Mixture of Experts [Shazeer et al., 2017], Hyper-Connections [Zhu et al., 2025], Neighborhood Attention [Hassani et al., 2023], and Masked Autoencoder pretraining [He et al., 2022] that we did not carry into the final model.

Positioning Otter. Previous work shows that skilful medium-range forecasting can be achieved at low cost with standard architectures. Otter goes further in three ways: it pushes the skill–compute frontier beyond these models, it ablates the individual design choices to identify which ones account for the gain, and it carries these choices into the probabilistic setting to obtain a highly compute-efficient probabilistic model.

3 Experimental Setup and Architectural Foundation

3.1 Task Specification

Following the well-established methodology for assessing medium-range AI weather models [Couairon et al., 2024, Nguyen et al., 2024, Lam et al., 2023], the task is to predict the temporal evolution of atmospheric variables using the ERA5 reanalysis dataset [Hersbach et al., 2020]. We predict future atmospheric states $X_{t+\delta t}$ at a lead time of $\delta t = 6$ hours, conditioned on a recent history of atmospheric states. The model predicts the residual difference relative to the most recent observation, and longer horizons are generated via autoregressive rollout. The state vector X_t comprises 87 variables: $AV = 6$ atmospheric variables X_t^{atm} (temperature, geopotential, specific humidity, vertical velocity, and U/V wind components) across 13 pressure levels, $SV = 4$ surface variables X_t^{surf} (2m temperature, mean sea-level pressure, and 10m U/V wind components), and 5 static topographic features. We train on data from 1979 to 2019 and evaluate on 2020, with forecasts initialised twice daily at 00:00 and 12:00 UTC. We provide additional architecture and training details in appendix B.

Spatial resolution. We conduct the majority of investigations at a spatial resolution of $1.5°$. While some SOTA models operate at $0.25°$, prior work [Allen et al., 2025, Couairon et al., 2024] establishes that $1.5°$ is a computationally efficient and reliable proxy for ablating architectural decisions, especially in the deterministic setting, where forecasting skill (i.e., RMSE) is primarily derived from low spatial frequency components. We test whether skill carries over to native resolution in section 4.3, where we scale Otter to $0.25°$ both by downscaling and by native training.

Evaluation Protocol. Following the standard WeatherBench 2 evaluation protocol [Rasp et al., 2024] at $1.5°$ (and $0.25°$, where applicable), we adopt established metrics and horizons suited to the respective modelling paradigms, separating the evaluation into deterministic and probabilistic. Furthermore, we distinguish between models evaluated with and without rollout fine-tuning (RFT)—a technique used to align the training objective with test-time behaviour, mitigating the train-test discrepancy for models that are initially trained with a one-step loss but evaluated via autoregressive rollouts. All metrics are averaged over the headline variables (Z500, Q700, T850, U850, V850, T2m, SP, U10m, and V10m) and over increments of 24h for a fair comparison to ArchesWeather which uses a 24h lead time.

- **Deterministic Evaluation.** We measure performance using the latitude-weighted Root Mean Squared Error (RMSE), mainly restricting our focus to lead times up to 72 hours. This is because models optimised via Mean Squared Error (MSE) target the conditional mean; at extended lead times, the considerable predictive uncertainty causes this averaging effect to produce unphysically smooth, blurred fields [Ben Bouallègue and the AIFS team, 2023]; we quantify this oversmoothing in appendix C.3.4. We divide this evaluation into two settings:
  - *Ablations (non-RFT):* To isolate architectural and optimisation choices from the confounding effects of RFT, we evaluate non-RFT models in our ablations. We report the average RMSE for 6–24h lead times—a window long enough to draw robust conclusions, yet short enough to avoid rewarding oversmoothing.
  - *Comparison to Baselines (RFT):* Because external SOTA baselines use different internal timesteps (e.g., 12 or 24 hours), we cannot apply the averaged evaluation protocol described above. For a fair comparison among autoregressive models, we uniformly align evaluation at a 24-hour lead time. We report the aggregated skill of our ***RFT*** models here, with extended evaluations up to 3 days provided in the Appendix.

- **Probabilistic Evaluation.** Because probabilistic models sample from a learned distribution rather than a smoothed mean, they enable meaningful evaluation at longer horizons. To obtain **Otter CRPS** and **Otter-XL CRPS**, the probabilistic models within the Otter family, we retrofit the deterministic backbone using a CRPS objective, followed by RFT. Performance is measured via the Continuous Ranked Probability Score (CRPS) applied to the rolled out model predictions up to 10 days. These CRPS scores are then averaged to provide a single headline number. Additionally, we evaluate a probabilistic deep ensemble constructed by applying diverse RFT hyperparameter configurations to the same base probabilistic checkpoint (**Otter(-XL) CRPS (Deep Ens)**).

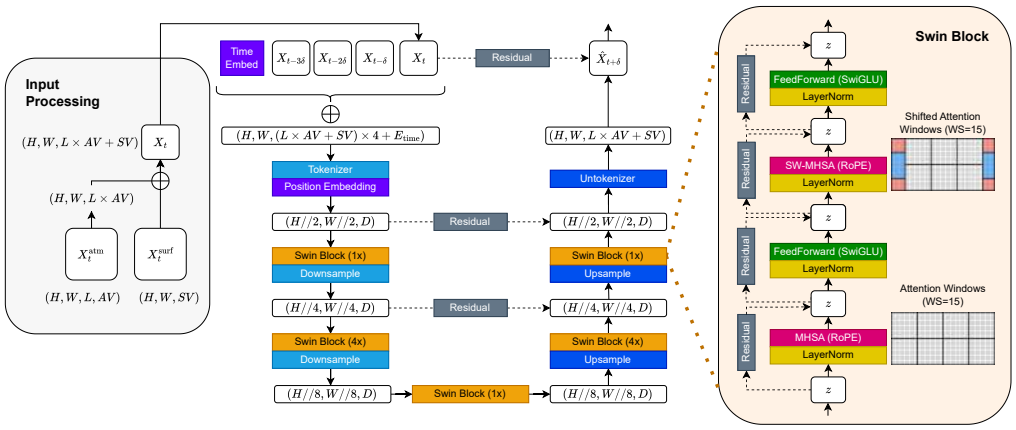


Figure 2: **Otter Weather architecture.** A 2D Swin Transformer predicts the future state $X_{t+\delta}$ from recent history $X_{t-(L-1)\times\delta}, \ldots, X_t$. Atmospheric variables across 13 pressure levels, surface variables, and time embeddings (E_{time}) are concatenated channel-wise at a spatial resolution of $H \times W = 121 \times 240$. Within the shifted-window attention (SW-MHSA) blocks, we employ RoPE and omit attention masks to enable cross-boundary communication, indicated by the shaded regions.

3.2 The Otter Base Architecture

Our design strictly minimises Earth-specific modifications, favouring standard, widely supported architectural components. This lets us directly use mature, low-level hardware optimisations and integrate architectural advancements proven across other AI domains, thereby maximising training efficiency. Furthermore, avoiding bespoke architectures lets the model be retrained on arbitrary variables without redesigning explicit pressure-level or variable-specific encoders.

Input Representation. Unlike architectures employing separate encoders for surface and upper-atmosphere variables [Lam et al., 2023, Couairon et al., 2024], we simply concatenate all input variables along the channel dimension (fig. 2). This tensor comprises SV surface variables, AV atmospheric variables, and 32 continuous multi-scale Fourier time embeddings. It is tokenised using a strided convolutional layer with patch size P and token dimensionality D.

Backbone. We use a 2D Swin Transformer [Liu et al., 2021] arranged in a symmetric U-Net configuration (Encoder, Bottleneck, Decoder). We deviate from the standard Swin in three ways: (1) we maintain a constant token dimensionality D across the U-Net; (2) we replace standard GeLU MLPs with SwiGLU feed-forward networks [Shazeer, 2020]; and (3) we replace relative position biases with 2D Rotary Positional Embeddings (RoPE) [Su et al., 2024]. Our Base configuration uses $[2, 8, 4]$ blocks per stage, 16 attention heads, and $D = 1536$.

Attention Masking and Topology. Standard models often apply custom attention masks to approximate a cylinder topology (masking the poles). This incurs significant compute overhead and breaks optimised attention kernels. Instead, we remove masking entirely, enabling communication across all boundaries (a torus topology) and relying on the positional embeddings to learn the spatial structure.

Optimisation and Regularisation. Otter is trained to minimise latitude- and pressure-weighted RMSE using the Muon optimiser [Jordan et al., 2024], which orthogonalises parameter updates to accelerate convergence. To offset Muon's matrix-operation overhead at small batch sizes, we use gradient accumulation. We apply strong regularisation typical of LLMs, including weight decay

(WD $= 0.15$), dropout ($p = 0.1$), and a batch-wise stochastic depth strategy (drop path probability 0.1), which skips residual blocks entirely for a mini-batch, improving both stability and training speed.

4 Pushing the Pareto Frontier in Weather Forecasting

4.1 Optimising the Deterministic Backbone

A critical advantage of Otter Weather's minimal hardware requirements is the rapid feedback loop: with training completing in under 3.5 A100-days, we can democratise architectural search. We systematically ablate key components to isolate high-impact design choices, evaluating their computational trade-offs averaged over 6-24h lead times (without RFT) to isolate architectural capacity.

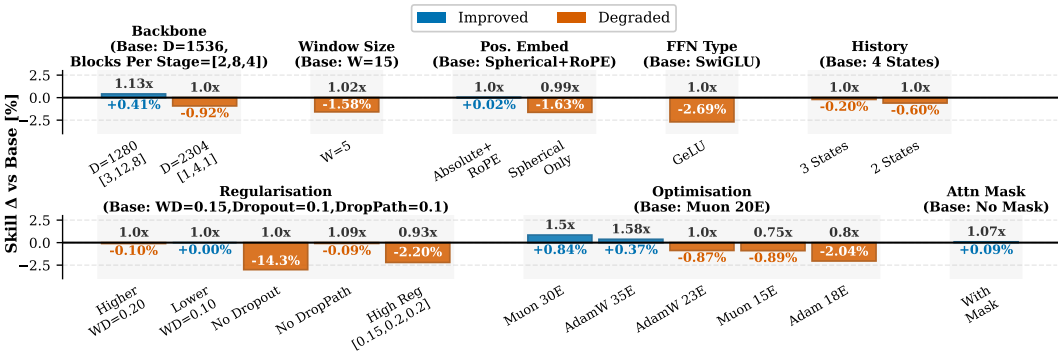


Figure 3: **Architectural and training ablations.** Bars show the relative percentage change in predictive skill (6–24h RMSE average of non-RFT models) versus the base model (black zero-line). Blue indicates improvement; orange denotes degradation. Multipliers above each bar reflect the relative computational cost, and labels of the form 20E give the number of training epochs. Deviations from the base configuration either degrade performance or incur disproportionate computational overheads that outweigh any marginal gains.

Ablation Insights. Our ablations (fig. 3) challenge several prevailing assumptions in weather modelling. Regarding the ***backbone*** profile, we found that a balanced, deeper U-Net configuration—distributing compute evenly across stages (e.g., $[2, 8, 4]$ blocks with $D = 1536$)— outperformed a "wide but shallow" setup ($[1, 4, 1]$ blocks with $D = 2304$) at a matched compute budget. We also found that optimising the ***FFN Type*** (i.e., replacing GeLU activations with SwiGLU) and the ***Pos. Embed*** (i.e., utilising 2D Rotary Positional Embeddings (RoPE)) yield consistent skill improvements. Surprisingly, omitting explicit attention masking (***Attn Mask***) across the Earth's poles (treating the topology as a torus rather than a cylinder) incurred negligible performance drops while drastically simplifying the implementation and providing a slight computational speedup. We further show in appendix C.3.3, by decomposing the effect of masking by latitude band, that the polar skill difference at 1.5° is not statistically significant. For ***optimisation***, the Muon optimiser consistently outperformed compute-matched AdamW across all evaluated training durations. Notably, the performance gap between the two widened as the number of epochs decreased, making Muon exceptionally advantageous for compute-constrained, rapid-iteration regimes. Effective ***regularisation*** proved vital, with dropout ($p = 0.10$) yielding the largest gains. We repeated the ablations in a different domain, an acoustic-scattering PDE task from The Well, and observed the same main trends (appendix D), which suggests the conclusions may hold more generally; with a single dataset, however, we treat this as preliminary evidence rather than a generalisation claim.

Evaluated and Excluded Components. Consistent with our goal of architectural simplicity and efficiency, we discarded several popular techniques from vision and language—including Hyper-connections, Neighborhood Attention (NATTEN), Masked Autoencoder (MAE) pre-training, and Mixture of Experts (MoE). While these are prominent in other domains, we found their marginal performance gains did not justify the added engineering complexity, dependency on custom kernels (in the case of NATTEN), or training fragility (for MoE). By documenting these ineffective directions, we aim to provide a transparent account of which modern advancements transfer readily to weather forecasting and which require significant tuning or customisation.

Deterministic Performance and Scaling. By combining these optimal, hardware-friendly components, the resulting **Otter Weather** base model extends the deterministic Pareto frontier (fig. 1a). After RFT, Otter achieves a **9.6% improvement** over the operational IFS HRES baseline across headline variables. Crucially, it surpasses the computationally heavy 4-member ArchesWeather ensemble (5.3% gain) while being more than six times more computationally efficient (<3.5 A100-days).

Furthermore, we demonstrate that our architecture scales predictably with increased compute. By reducing the patch size from 2 to 1, we train **Otter-XL** for approximately 10 A100-days, achieving a **14.9% gain** (fig. 1a) over IFS HRES at a 24h lead time. This result places it on par with the industrial-scale GraphCast (14.6% gain) when evaluated at 1.5°, effectively matching SOTA performance despite requiring over two orders of magnitude less compute. We present in fig. 9 the evolution of the RMSE with lead time, showing that Otter-XL maintains competitive performance even up to a 10-day lead time horizon across the majority of variables.

4.2 Probabilistic Otter Weather

While our deterministic model is highly efficient, the weather forecasting community has increasingly shifted towards probabilistic paradigms to capture predictive uncertainty. We demonstrate that Otter's architectural efficiency transfers to this setting, allowing us to also push the probabilistic skill-compute Pareto frontier. To achieve this, we evaluate two distinct pipelines: training a probabilistic model from scratch using the CRPS objective (**Otter CRPS (scratch)**) and fine-tuning an existing deterministic backbone via CRPS (**Otter CRPS**) [Diaconu et al., 2026].

Deep Ensembles through RFT. To construct our deep ensembles, we leverage the RFT phase (figs. 6 and 7). This approach differs from the concurrent work U-Cast [Cachay et al., 2026], which achieves deep ensembling by training multiple one-step CRPS fine-tuned models from different random seeds. By pushing the branching point to the RFT stage, we can efficiently generate deep ensembles using either of our two methodologies (deterministic pre-training $\rightarrow$ CRPS fine-tuning $\rightarrow$ multiple RFTs, or CRPS from-scratch $\rightarrow$ multiple RFTs). This is a critical advantage: it makes from-scratch deep ensembling computationally viable, avoiding the $>$ 20 A100-day cost per model seen in other architectures, and ensures practitioners retain full, independent control over the entire pre-training pipeline.

Furthermore, the ensemble members come at no additional cost beyond the standard development cycle. Rather than training with different random seeds, we construct the ensemble using the checkpoints generated during standard RFT hyperparameter tuning (specifically, by varying the learning rates), thus bypassing the need for full-scale retraining.[2] To evaluate what this saving costs in skill, we compare against a conventional deep ensemble of models trained with different random seeds (RS Deep Ens in fig. 4). The seed ensemble is modestly more skilful (6.4% vs 6.1% CRPS improvement), so reusing sweep checkpoints costs a small amount of skill in exchange for avoiding the additional training runs. Both variants show that effective ensembles can be built under tight resource constraints; identifying the optimal branching point in the training pipeline for ensemble diversity and calibration remains a direction for future work.

CRPS Objective. We optimise the probabilistic version of Otter Weather with the Continuous Ranked Probability Score (CRPS) Gneiting and Raftery [2007], a proper scoring rule applied to the marginal distributions of the output. We generate an ensemble of M forecasts $(X_1, \cdots X_M)$ for a given ground-truth observation y and compute the unbiased empirical CRPS estimator ('fair' CRPS following Alet et al. [2025]:

$$\text{CRPS}_{\text{fair}} = \frac{1}{M}\sum_{i=1}^{M}|X_i - y| - \frac{1}{2M(M-1)}\sum_{i=1}^{M}\sum_{j=1}^{M}|X_i - X_j|, \tag{1}$$

where the first term encourages each ensemble member to be accurate (and minimise the absolute error), while the second term encourages dispersion in the ensemble members.

Injecting Stochasticity: AdaLN vs. MC Dropout. Some source of stochasticity needs to be used to generate the M ensemble members X_i. The prevailing approach in weather and diffusion models is to inject latent noise via Adaptive Layer Normalisation (AdaLN) [Lang et al., 2026, Alet et al., 2025, Diaconu et al., 2026]. However, drawing inspiration from concurrent findings in U-Net architectures [Cachay et al., 2026], we ablate this standard AdaLN approach against a simpler,

[2]We note that we depart from the standard definition of deep ensembles from Lakshminarayanan et al. [2017], and employ a method that shares conceptual similarities to Garipov et al. [2018]. We refer to appendix B.3.5 for a more detailed comparison.

parameter-free alternative: Monte Carlo (MC) Dropout. In our Transformer-based setting, we find that MC Dropout naturally aligns with the existing dropout layers used during deterministic pre-training. Our ablations reveal that MC Dropout not only avoids the parameter overhead of AdaLN blocks but actually achieves superior rollout CRPS metrics, though it results in slightly more underdispersed ensembles, as indicated in fig. 4.

Regarding the dropout ablations, the Muon optimiser proved essential for convergence, consistently outperforming AdamW in rollout CRPS (see fig. 11), which aligns with findings from U-Cast. However, Muon-optimised ensembles are slightly more underdispersed. Ablations of the AdaLN injection strategy (fig. 12) reveal that the backbone learning rate is the primary driver of calibration (measured through the spread-skill ratio (SSR)), while the noise branch learning rate significantly influences the skill-spread trade-off. Generally, lower learning rates favour calibration over raw CRPS skill. Furthermore, increasing the noise embedding dimension D beyond 64 yielded negligible improvements while increasing parameter count and training latency, and higher weight decay (0.2) on the noise branch proved strictly inferior to 0.1 for both skill and calibration.

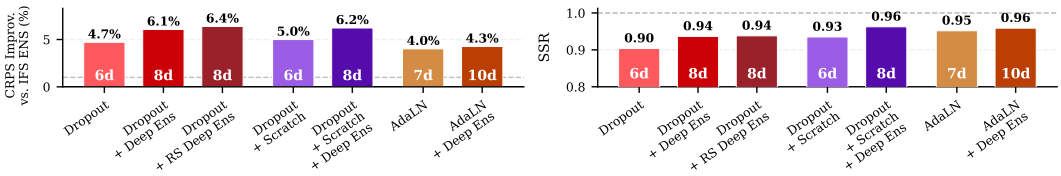
From-scratch versus fine-tuning. We additionally compare Otter CRPS (fine-tuned from the deterministic checkpoint) to an equivalent model trained from scratch under a matched compute budget—namely, the sum of the deterministic pretraining and CRPS fine-tuning costs. While the final 6h training loss of the model trained from scratch is higher than that of the fine-tuned model (7% higher—0.29 versus 0.27), after the RFT procedure we surprisingly find that both variants achieve similar performance, with the model trained from scratch achieving a 6.2% average CRPS improvement over 1-10 days lead times as compared to 6.1% for the fine-tuned model (fig. 4). This suggests that, with our efficient architecture, both options are viable choices for obtaining a competitive probabilistic model in less than 8 A100 days, with the latter preferred if the deterministic checkpoint is already available, or when practitioners want to maintain control over the pre-training pipeline.

Figures 13 and 14 show the evolution of the CRPS improvement over IFS ENS and of the SSR with lead time up to 10 days, showing that the two methods achieve very similar CRPS metrics, but the models trained from scratch tend to show better calibration profiles—while Otter CRPS generally tends to be underdispersed, especially at the shorter lead times, the SSR becomes closer to 1 for the model trained from scratch versus the fine-tuned one. Example predictions for both probabilistic variants are available in appendix C.6.

Deep Ensemble: learning rates vs. random seeds. The standard practice for constructing deep ensembles relies on training multiple independent models with distinct random seeds [Lakshminarayanan et al., 2017]. As illustrated in fig. 4, we compare this conventional approach against our efficiency-driven method, which constructs an ensemble through model checkpoints generated during standard learning rate tuning. Under a computational budget of 8 A100-days, our method yields a highly competitive 6.1% Continuous Ranked Probability Score (CRPS) improvement over the IFS ENS baseline and maintains a Spread-Skill Ratio (SSR) of 0.94. When employing the standard random seed approach (RS Deep Ens) within the exact same compute footprint, we observe a modest performance advantage, pushing the CRPS improvement to 6.4% while maintaining the 0.94 SSR. These results indicate that while explicitly varying random seeds offers slightly superior predictive skill, repurposing hyperparameter checkpoints successfully produces a calibrated, highly performant ensemble in the low-compute regime without requiring dedicated retraining runs.

Probabilistic Pareto Frontier. By combining MC Dropout with an RFT ensembling strategy, we achieve an effective probabilistic medium-range weather model at a fraction of the cost of current SOTA models. As demonstrated in fig. 1b, this approach fundamentally advances the probabilistic skill-compute Pareto frontier. Using a total budget of under 8 A100-days, the base Otter model surpasses flow-based baselines like ArchesWeatherGen while requiring less than half of their > 20 A100-day training footprint. When scaling this framework to larger compute regimes, the underlying architecture remains exceptionally performant. As mapped out in the skill-compute frontier of fig. 1b (and detailed across model configurations with Otter family in fig. 5), our scaled variant, Otter-XL, dominates existing baselines. In particular, Otter-XL (Deep Ens) proves remarkably capable: requiring only 29 A100-days of training compute, it achieves a 9.7% CRPS improvement over the IFS ENS baseline while maintaining a Spread-Skill Ratio (SSR) of 0.95. This allows the model to deliver a nearly two-fold skill increase over comparable lightweight architectures, while simultaneously outperforming resource-intensive frontier models like GenCast by a clear 2.2% margin at a 10-fold reduction in training compute. In figs. 15 and 16, we include Otter XL in the lead-time evolution of

CRPS improvement over IFS ENS and SSR up to 10 days, demonstrating that the gains from a larger model persist across all lead times, though they diminish as lead time increases.



(a) CRPS (b) SSR

Figure 4: **Performance of Probabilistic Training Strategies.** Average CRPS improvement over the IFS ensemble (a) and SSR (b) across 1–10 day lead times for three Otter CRPS variants: fine-tuning with MC Dropout (red bars), from-scratch training (purple bars), and fine-tuning with AdaLN injection (yellow/orange bars). Numbers inside the bars indicate training cost in A100-days. For each base configuration, employing a Deep Ensemble (Deep Ens) consistently improves both skill and spread.

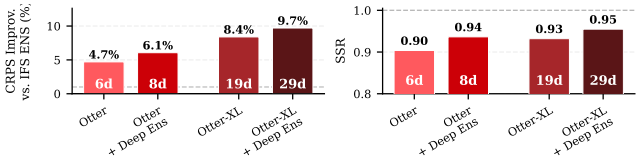


Figure 5: **Scaling behaviour of probabilistic Otter models.** CRPS improvement over IFS ENS (%, left) and SSR (right) across 1–10 day lead times for single-model and Deep Ens configurations of Otter and Otter-XL. Compute costs are annotated inside the bars in A100-days. At $\sim 3.5\times$ the compute, Otter-XL yields over a 55% relative increase in skill over the corresponding base configuration.

Table 1: **Skill at** $0.25°$**.** Mean RMSE improvement (%) over IFS HRES across the nine headline variables, 2020. Otter-dwn: Otter-XL with the post-hoc ridge downscaler; Otter-0.25: trained natively (17 A100-days). Best per column in bold.

Model	24 h	72 h	240 h
Otter-dwn	+14.6	+17.1	**+10.7**
Otter-0.25	+17.3	+15.0	+7.6
GraphCast	**+19.6**	**+17.7**	+9.9

4.3 Scaling to $0.25°$

To test whether Otter's skill holds at ERA5's native resolution, we bring the deterministic Otter-XL to $0.25°$ by two routes. **Post-hoc downscaling (Otter-dwn)** fits a per-grid-cell ridge regression from a $1.5°$ neighbourhood of the forecast to its residual over bilinear interpolation at $0.25°$. We train on ERA5 over 2016–2019 and apply the downscaler to Otter-XL's forecasts at inference. The additional training cost of this strategy is negligible, and we can obtain downscaled predictions from any $1.5°$ model. **Native training (Otter-0.25)** trains Otter from scratch on $0.25°$ data with patch size 6, which gives the same token count as Otter-XL at $1.5°$ and leaves the backbone unchanged; we follow the same recipe as in section 4.1, followed by RFT, for a total training time of 17 A100-days.

Table 1 reports mean skill over the nine headline variables against GraphCast; per-variable results and a comparison to FuXi are in appendix C.7. Both routes reach competitive skill but distribute it differently. Otter-dwn trails GraphCast at 24 h, where fine-scale detail matters and a linear downscaler without orography cannot recover it, but is within 0.6 points at 72 h and ahead thereafter. Otter-0.25 shows the opposite profile: it beats the downscaler on eight of nine variables at 24 h, but falls 2–3 points behind GraphCast at longer horizons, where the outcome depends on an RFT procedure we did not tune for this resolution.

5 Conclusion

We introduced Otter Weather to demonstrate that, at $1.5°$ on ERA5, competitive global weather forecasting requires neither massive compute budgets nor domain-specific inductive biases. By systematically integrating general-purpose advances from the LLM and vision communities, we advance the skill-compute Pareto frontier for both deterministic and probabilistic modelling. Importantly, we achieve these gains operating under a highly constrained training budget: under 3.5 A100-days for our deterministic model and under 8 A100-days for our probabilistic deep ensembles. Finally, we

show that the deterministic model yields competitive performance at 0.25° at a small fraction of the cost of existing high-resolution models. This removes two dependencies at once: on industry-scale compute, and on design decisions within released foundation models that practitioners currently inherit whether or not they suit their problem.

Otter Weather provides a powerful template for democratised and sovereign weather modelling, filling a critical gap in the ecosystem of data-driven forecasting. A healthy landscape must include not only expensive, general-purpose foundation models, but also lightweight, cheaply trainable alternatives. This diversity is essential for two complementary reasons. First, true scientific sovereignty requires that forecasting tools be adaptable to specific institutional needs and compute budgets without relying solely on large-scale, pre-trained models. Second, operational forecasting has long relied on ensembles of distinct models; by aggregating systems with different failure modes, we can effectively reduce correlated biases and epistemic uncertainty. Because data-driven models dramatically lower inference costs compared to traditional numerical methods, deploying these large, varied ensembles is now increasingly practical, further amplifying the need for the architectural and training diversity that accessible models like Otter provide.

Limitations and Future Work. While Otter achieves competitive performance, our approach is not without limitations (detailed in appendix E). Our findings are empirical and were mainly obtained at 1.5°; only the deterministic variant has been scaled to 0.25°, and domain-specific priors may matter more at that resolution or for physical consistency. Secondly, like many data-driven methods, Otter offers fewer ways for physical interpretation as opposed to models with explicit structure. Additionally, while our deep ensembles yield significant skill improvements, the optimal method for constructing them remains an exciting open question. Future work should systematically evaluate the trade-offs between training completely independent models from scratch versus branching ensemble members from a shared pre-trained checkpoint. Despite these limitations, Otter Weather provides a highly accessible, robust foundation for future research, aimed at democratising weather forecasting.

Reproducibility Statement

Detailed experimental settings are provided in the Appendix. To facilitate reproducibility, the complete implementation is available at https://github.com/cambridge-mlg/otter.

Acknowledgements

This work was funded by the EPSRC Probabilistic AI Hub (EP/Y028783/1), the Horizon Europe grant 101213369 (DVPS), and a Microsoft AI4Good Azure credits donation. Cristiana Diaconu is supported by the Cambridge Trust Scholarship. Jonas Scholz is supported by the Cambridge Zero/Marshall Foundation Scholarship. We would also like to acknowledge the support of the Simons Foundation and of Schmidt Sciences. This work was supported in part by the AI2050 program at Schmidt Sciences (Grant G-25-70028).

References

Ferran Alet, Ilan Price, Andrew El-Kadi, Dominic Masters, Stratis Markou, Tom R. Andersson, Jacklynn Stott, Remi Lam, Matthew Willson, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Skillful joint probabilistic weather forecasting from marginals, 2025. URL https://arxiv.org/abs/2506.10772.

Anna Allen, Stratis Markou, Will Tebbutt, James Requeima, Wessel P. Bruinsma, Tom R. Andersson, Michael Herzog, Nicholas D. Lane, Matthew Chantry, J. Scott Hosking, and Richard E. Turner. End-to-end data-driven weather prediction. *Nature*, 641(8065):1172–1179, 2025. doi: 10.1038/s41586-025-08897-0.

Zied Ben Bouallègue and the AIFS team. A new ML model in the ECMWF web charts. ECMWF AIFS Blog, 2023. URL https://www.ecmwf.int/en/about/media-centre/aifs-blog/2023/new-ml-model-ecmwf-web-charts.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970):533–538, 2023. doi: 10.1038/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, 2025. doi: 10.1038/s41586-025-09005-y.

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

Salva Rühling Cachay, Miika Aittala, Karsten Kreis, Noah D Brenowitz, Arash Vahdat, Morteza Mardani, and Rose Yu. Elucidated rolling diffusion models for probabilistic forecasting of complex dynamics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=NnRQOPzL9P.

Salva Rühling Cachay, Duncan Watson-Parris, and Rose Yu. U-Cast: A surprisingly simple and efficient frontier probabilistic AI weather forecaster. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=XF0wkyEbuM.

Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. FuXi: a cascade machine learning forecasting system for 15-day global weather forecast. *npj Climate and Atmospheric Science*, 6:190, 2023. doi: 10.1038/s41612-023-00512-1.

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

Cristiana Diaconu, Miles Cranmer, Richard E. Turner, Tanya Marwah, and Payel Mukhopadhyay. Probabilistic retrofitting of learned simulators. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, pages 24654–24701. PMLR, 2026. URL https://proceedings.mlr.press/v306/diaconu26a.html.

Dan Fu, Hermann Kumbong, Eric Nguyen, and Christopher Ré. FlashFFTConv: Efficient convolutions for long sequences with tensor cores. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 9455–9483, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/281190d87732639e63bc19dedd7d711b-Paper-Conference.pdf.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing*

Systems, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/be3087e74e9100d4bc4c6268cdbe8456-Paper.pdf.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6185–6194, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 15979–15988. IEEE, 2022.

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. doi: 10.1002/qj.3803.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024. Accessed: 2026-02-03.

Emma Kasteleyn, Timo Maier, Axel Lauer, Veronika Eyring, Pierre Gentine, and Ana Lucic. PhysMetrics.Weather: An evaluation framework for physical consistency in ML weather models, 2026. URL https://arxiv.org/abs/2606.10642.

Ryan Keisler. Forecasting global weather with graph neural networks, 2022. URL https://arxiv.org/abs/2202.07575.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, Sam Hatfield, Peter Battaglia, Alvaro Sanchez-Gonzalez, Matthew Willson, Michael P. Brenner, and Stephan Hoyer. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024. doi: 10.1038/s41586-024-07744-y.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. doi: 10.1126/science.adi2336.

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, 2024. URL https://arxiv.org/abs/2406.01465.

Simon Lang, Mihai Alexe, Mariana C. A. Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D. Dueben, Sara Hahner, Pedro Maciel, Ana Prieto-Nemesio, Cathal O'Brien, Florian Pinault, Jan Polster, Baudouin Raoult, Steffen Tietsche, and Martin Leutbecher. AIFS-CRPS: ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *npj Artificial Intelligence*, 2026. doi: 10.1038/s44387-026-00073-7.

Fanny Lehmann, Firat Ozdemir, Benedikt Soja, Torsten Hoefler, Siddhartha Mishra, and Sebastian Schemm. Finetuning a weather foundation model with lightweight decoders for unseen physical processes, 2025. URL https://arxiv.org/abs/2506.19088.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training, 2025. URL https://arxiv.org/abs/2502.16982.

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)*, pages 9992–10002, 2021. doi: 10.1109/ICCV48922.2021.00986.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ankur Mahesh, William D. Collins, Boris Bonev, Noah Brenowitz, Yair Cohen, Joshua Elms, Peter Harrington, Karthik Kashinath, Thorsten Kurth, Joshua North, Travis O'Brien, Michael Pritchard, David Pruitt, Mark Risser, Shashank Subramanian, and Jared Willard. Huge ensembles – Part 1: Design of ensemble weather forecasts using spherical Fourier neural operators. *Geoscientific Model Development*, 18(17):5575–5603, 2025. doi: 10.5194/gmd-18-5575-2025.

Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Régaldo-Saint Blancard, François Rozet, Cristiana Diaconu, Lucas Thibaut Meyer, Kaze W. K. Wong, Mohammad-Hadi Sotoudeh, Alberto Bietti, Irina Espejo Morales, Rio Alexa Fear, Siavash Golkar, Tom Hehir, Keiya Hirashima, Geraud Krawezik, François Lanusse, Rudy Morel, Ruben Ohana, Liam Holden Parker, Mariel Pettee, Jeff Shen, Kyunghyun Cho, Miles Cranmer, and Shirley Ho. Walrus: A cross-domain foundation model for continuum dynamics. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=kcyOjXoUZu.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. ClimaX: A foundation model for weather and climate. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 25904–25938. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/nguyen23a.html.

Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcomano, Romit Maulik, Veerabhadra Kotamarthi, Ian Foster, Sandeep Madireddy, and Aditya Grover. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pages 68740–68771, 2024. doi: 10.52202/079017-2196.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Keaton Burns, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, pages 44989–45037, 2024. doi: 10.52202/079017-1430.

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, 2022. URL https://arxiv.org/abs/2202.11214.

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. doi: 10.1038/s41586-024-08252-9.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallègue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024. doi: 10.1029/2023MS004019. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2023MS004019.

Marc Rußwurm, Konstantin Klemmer, Esther Rolf, Robin Zbinden, and Devis Tuia. Geographic location encoding with spherical harmonics and sinusoidal representation networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PudduufFLa.

Noam Shazeer. GLU variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Christopher Subich. Efficient fine-tuning of 37-level GraphCast with the Canadian global deterministic analysis. *Artificial Intelligence for the Earth Systems*, 4(3), July 2025. ISSN 2769-7525. doi: 10.1175/AIES-D-24-0101.1.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9FqARW7dwB.

A Summary Table of Related Work

We provide a comparative summary of deterministic weather models in table 2. Despite operating with the lowest compute budget, Otter Weather achieves superior performance in the medium-resolution category and remains competitive with state-of-the-art models in the high-resolution regime. Otter-XL closely approaches the performance of the best high-resolution model (GraphCast) despite using two orders of magnitude less compute.

Table 2: **Comparison of deterministic global weather models. Params** refers to the total number of parameters. **Cost** refers to training compute in A100-day equivalents (adapted from Couairon et al. [2024]). **Skill** is average RMSE improvement (%) over IFS HRES, aggregated over 24h–72h (↑).

Model	Architecture	Res.	Params	Cost	Skill %
Numerical models					
IFS HRES	Numerical	0.1°	–	–	0.00
High-resolution (< 1°)					
Pangu-Weather [Bi et al., 2023]	3D Earth-Specific Transformer	0.25°	256M	1440	5.84
FuXi [Chen et al., 2023]	Cascade U-Transformer	0.25°	1.5B	**26**	14.44
NeuralGCM (0.7°) [Kochkov et al., 2024]	Hybrid (Diff. Dynamics + NN)	0.7°	31.1M	8064	14.61
GraphCast [Lam et al., 2023]	Multi-mesh GNN	0.25°	36.7M	1344	**16.32**
Medium-resolution (> 1°)					
Keisler GNN [Keisler, 2022]	Icosahedral GNN	1.0°	6.7M	5.5	-15.27
Stormer ENS (mean) [Nguyen et al., 2024]	Vision Transformer (ensemble)	1.5°	300M	128	6.11
NeuralGCM 50×ENS [Kochkov et al., 2024]	Hybrid (Diff. Dynamics + NN)	1.4°	18.3M	7680	7.79
ArchesWeather-M×4 [Couairon et al., 2024]	Swin U-Net (ensemble)	1.5°	84M	20	8.86
Otter Weather (Ours)	**Constant-Width 2D Swin U-Net**	1.5°	1.5B	**3**	11.54
Otter-XL (Ours)	**Constant-Width 2D Swin U-Net**	1.5°	1.5B	10	16.12

B Detailed Architecture and Training Details

This appendix provides a detailed specification of our deterministic model architecture.

B.1 Input Representation

Normalisation and Flattening. Conditioned on L historical states, the input tensor to the network has dimensions (B, L, H, W, C), where B denotes the batch size, L the temporal context length, $H = 121$ and $W = 240$ the spatial resolution (corresponding to a 1.5° grid), and $C = 87$ the number of modelled variables. We first apply standard normalisation using variable-specific statistics (mean and standard deviation) computed over the training corpus. Subsequently, we flatten the temporal and channel dimensions to obtain a tensor of shape $(B, H, W, L \cdot C)$. This reshaping allows the 2D backbone to process temporal history as distinct input channels, enabling the learning of spatial-temporal correlations through channel mixing mechanisms.

Temporal Embeddings. Atmospheric dynamics are characterised by phenomena occurring across a wide range of temporal scales, many of which exhibit periodic behaviour. To capture these cyclic dependencies, we follow Bodnar et al. 2025 and generate Fourier-based temporal embeddings derived from the time elapsed t (in hours) relative to a fixed reference date (January 1, 1979). Specifically, for a set of $N = 16$ scales s_i spaced logarithmically between ζ_{min} and ζ_{max}, we compute the embedding $\text{Emb}(t)$ as:

$$\text{Emb}(t) = \bigoplus_{i=1}^{N} \left[\sin\left(\frac{2\pi t}{s_i}\right), \cos\left(\frac{2\pi t}{s_i}\right) \right]. \tag{2}$$

We select $\zeta_{\text{time,min}} = 3.0$ and $\zeta_{\text{time,max}} = 8760$ (corresponding to the number of hours in a year). As argued in Bodnar et al. 2025, this multi-scale embedding allows the model to resolve critical temporal signals, such as diurnal cycles, weekly patterns, and seasonal variations.

Residual Learning. Consistent with recent advances in the field [Lam et al., 2023, Nguyen et al., 2024], we formulate the prediction task as learning the residual update (tendency) rather than the full future state. The prediction for the next step is defined as an additive update to the most recent context state X_t:

$$\hat{X}_{t+\Delta t} = X_t + \text{Backbone}(X_{t-T:t}, \text{Emb}(t + \Delta t)).$$

B.2 Backbone: Swin U-Net.

The core backbone of our model is a Swin Transformer [Liu et al., 2021] adapted into a symmetric encoder-decoder (U-Net) architecture. This hierarchical design allows the model to capture multi-scale atmospheric phenomena.

B.2.1 Patch Partitioning (Tokeniser).

The first step in the backbone is to transform the input tensor into a sequence of discrete tokens. We implement this using a Patch Partition layer, implemented as a strided convolution. Given an input tensor of shape $(B, H, W, L \cdot C)$, we project local patches into a latent embedding dimension D. By setting the stride P equal to the patch size, we partition the image into a grid of non-overlapping windows, resulting in a representation of shape $(B, H/P, W/P, D)$.

B.2.2 Positional Embeddings.

We explore multiple strategies to inject positional information into the architecture.

Absolute Fourier Spatial Embeddings. We follow the strategy employed in Bodnar et al. 2025 to encode absolute positional information through Fourier Embeddings. Given a number $N_s = 128$ of

[2]We are reporting the parameter count for Lang et al. 2026 (the probabilistic version of the model) as we could not find the corresponding count for the deterministic model.

scales, the first $N_s/2$ are used for latitude (ϕ) information, and the next $N_s/2$ for longitude (λ). Each coordinate is independently projected into Fourier features as described in appendix B.1. We use $\zeta'_{\text{pos,min}} = 0.1$ and $\zeta'_{\text{pos,max}} = 720$. These feature vectors are broadcast to form 2D spatial grids and concatenated along the channel dimension. Finally, a learnable linear layer projects the concatenated features to the model's token dimension D before adding them to the input tokens:

$$E_{pos} = \text{Linear}(\text{Emb}(\phi) \oplus \text{Emb}(\lambda))$$

Spherical Harmonics Embedding. To contrast the use of standard positional embeddings with more specialised versions that strictly respect spherical symmetry, we also investigate the use of spherical harmonics embeddings following Rußwurm et al. 2024. We pre-compute the real spherical harmonic basis functions $Y_\ell^m(\theta, \phi)$ up to a maximum degree $L_{\text{max}} = 20$. For every spatial location (h, w), we generate a dense feature vector containing all harmonic terms for degrees $0 \leq \ell \leq L_{\text{max}}$ and orders $-\ell \leq m \leq \ell$. This results in $(L_{\text{max}} + 1)^2 = 441$ unique geometric features per pixel. These raw harmonic features are cached and passed through a learnable linear projection to match the token dimension D before being added to the input tokens.

Rotary Positional Embeddings (RoPE). While absolute embeddings provide global context, we also investigate the use of relative positional embeddings. We employ Rotary Positional Embeddings (RoPE) [Su et al., 2024] within the self-attention mechanism of each Swin block.

We use a 2D extension of RoPE where the spatial indices (i, j) of a token rotate the query and key vectors in the complex plane. This allows the attention mechanism to depend only on the relative distance $\Delta = (i_1 - i_2, j_1 - j_2)$ between tokens.

B.2.3 Hierarchical Encoder-Decoder Structure

The processed tokens pass through a symmetric U-Net structure composed of Swin Transformer Stages.

We adopt a "constant-width" U-Net configuration, maintaining the same channel capacity across all resolutions. The network consists of 2 downsampling levels (3 stages total, including the bottleneck). The hyperparameters for the base configuration are:

- Embedding Dimension (D): 1536.
- Encoder Depth: 2 Stages with block counts $[2, 8]$.
- Bottleneck Depth: 4 Blocks.
- Decoder Depth: 2 Stages with block counts $[8, 2]$ (symmetric to encoder).
- Attention Heads: 16 heads per block (dimension per head $d_k = 1536/16 = 96$).
- Window Size: 15×15.

This results in a symmetric depth profile of (2, 8, 4, 8, 2) blocks. The downsampling operations reduce the spatial resolution by a factor of 2 at each stage (via 2×2 strided convolution) while maintaining the channel dimension constant at D. The skip connections use complex fusion (concatenation followed by a 1×1 convolution) rather than simple summation.

B.2.4 Swin Transformer Block and Feed-Forward Network

The fundamental building block of our architecture is the Swin Transformer Block [Liu et al., 2021]. To balance local feature extraction with long-range dependencies, each block consists of two attention operations:

- **Window Attention (W-MSA).** The first unit partitions the input state into non-overlapping local windows of size $W \times W$ (where $W = 15$ in the base configuration) and computes self-attention independently within each window.
- **Shifted Window Attention (SW-MSA).** The second unit enables cross-window information exchange by shifting the window partitioning by $(\lfloor \frac{W}{2} \rfloor, \lfloor \frac{W}{2} \rfloor)$ before computing self-attention.

SwiGLU Feed-Forward Network. We replace the standard GELU-MLP used in the original Swin architecture with a SwiGLU variant, a modification shown to improve stability in large-scale models [Shazeer, 2020, Couairon et al., 2024]. The activation is defined as:

$$\text{SwiGLU}(x) = (\text{SiLU}(xW_1) \otimes xW_2)W_3$$

where $\otimes$ denotes element-wise multiplication. To maintain parameter parity with standard Transformer MLPs (typically expansion ratio 4), we reduce the hidden dimension expansion to $\approx \frac{8}{3}$ ($2.67\times$). Additionally, we enforce hardware-aware alignment by padding the hidden dimension to the nearest multiple of 128, ensuring optimal memory tiling for GPU Tensor Cores.

B.3 Additional Training Details

B.3.1 Loss.

The deterministic model is trained to minimise a weighted sum of per-channel, latitude-weighted Root Mean Squared Errors (RMSE). For a variable v at pressure level ℓ and lead time t, the per-channel loss is

$$\mathcal{L}^t_{v,\ell} = \sqrt{\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} w(\phi_i)\left(\hat{X}^t_{i,j,v,\ell} - X^t_{i,j,v,\ell}\right)^2}, \qquad w(\phi_i) = \frac{\cos\phi_i}{\frac{1}{H}\sum_{k=1}^{H}\cos\phi_k}, \tag{3}$$

where ϕ_i is the latitude of row i on the uniform $H \times W$ grid, so that each cell contributes in proportion to its area. Channels are weighted by

$$w_{v,\ell} = \left(\frac{\ell}{\sum_{\ell' \in \mathcal{L}} \ell'}\right)^p s_v^p \, \sigma_{v,\ell}^{-p}, \tag{4}$$

where $\mathcal{L} = \{50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, 1000\}$ hPa are the 13 pressure levels, $\sum_{\ell'} \ell' = 6025$, and $p = 1$ in all reported runs. The level component $\ell/6025$ is linear in pressure and sums to one across the 13 levels; surface variables take a level component of 1. Following Lam et al. [2023], Couairon et al. [2024], the per-variable factor is $s_v = 1$ for atmospheric variables and 2m temperature and $s_v = 0.1$ for 10m winds and mean sea-level pressure. $\sigma_{v,\ell}$ is the standard deviation of the field, pooled over latitude, longitude and time, so that channels are compared in normalised units. The total loss sums over channels and averages over lead times $t = 1, \ldots, T$:

$$\mathcal{L} = \sum_{v,\ell} w_{v,\ell} \, \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}^t_{v,\ell}. \tag{5}$$

For CRPS training, we apply the same weighting to the 'fair' CRPS (eq. (1)), evaluated during training with $M = 2$ ensemble members.

B.3.2 Optimisation.

We train Otter Weather using the Muon optimiser [Jordan et al., 2024]. By treating 2D parameters natively as matrix objects and transforming the momentum from both sides, Muon orthogonalises the updates—encouraging learning along non-dominant directions and accelerating convergence [Liu et al., 2025]. In our setting, however, this matrix-level operation makes each Muon step substantially more expensive than an AdamW step, a gap amplified by the small batch sizes typical in weather modelling. We offset this overhead by using gradient accumulation (4 mini-batches per optimisation step), which increases the effective batch size and amortises Muon's per-step cost. With this modification, we find that Muon improves training convergence. Notably, we observe that while the *presence* of learning rate decay is critical for performance, the specific decay profile (e.g., linear vs. cosine) is of less importance.

B.3.3 Hardware & Compute Budget.

We detail below the hardware used in this investigation. Our cluster includes nodes equipped with:

- NVIDIA RTX 6000 (Blackwell Server Edition), 128 vCPUs, 96GB VRAM;

- NVIDIA A100-PCIe (80GB), 24 vCPUs, 80GB VRAM;
- NVIDIA RTX 6000 (Ada Generation), 28 vCPUs, 48GB VRAM.

To ensure fair comparison and reproducibility, all computational costs reported in the main text are normalised to the equivalent of A100-80GB GPU hours. Furthermore, regardless of the node capacity, all experiments are strictly conducted in a single-GPU regime, ensuring that our results are representative of performance achievable on accessible hardware without requiring multi-GPU distributed training strategies. Peak VRAM achieved during training of the models using the deterministic Base configuration was 46.5GB, and inference time for one autoregressive step takes approximately 0.48s.

B.3.4 Learning Rate Schedule.

Deterministic model. We employ a cosine decay learning rate schedule with a linear warmup of 500 steps.

- **Pretraining:** The learning rate increases linearly to a maximum of 2×10^{-4} during warmup, followed by a cosine decay to a minimum of 2×10^{-6}.
- **Rollout Fine-tuning (RFT):** To ensure stability, we initialise the fine-tuning schedule at the final learning rate of the pretraining phase and decay to 0 following a cosine schedule.

Probabilistic model: Otter CRPS.

- **One-step CRPS pretraining:** Starting from the pretrained deterministic base Otter model, we employ a cosine decay learning rate schedule with a linear warmup of 500 steps. The learning rate increases linearly to a maximum of 2×10^{-4} during warmup, followed by a cosine decay to a minimum of 2×10^{-6}.
- **Rollout Fine-tuning (RFT):** For this stage, we employ a cosine decay learning rate schedule without a warmup, decaying to a minimum of 10^{-9}. For the best single **Otter CRPS** model, we used an initial learning rate of 5×10^{-6}. To create the **Otter CRPS (Deep Ens)**, we combine three models fine-tuned with initial learning rates of 10^{-6}, 2×10^{-6}, and 5×10^{-6}.

Probabilistic model: Otter CRPS (scratch).

The **Otter CRPS (scratch)** model is initialized from scratch rather than the pretrained base. It utilises the exact same learning rate schedules for both the one-step CRPS pretraining and Rollout Fine-tuning stages as described above, differing only in the number of epochs for which the models are trained.

Probabilistic model: Otter CRPS with AdaLN.

- **One-step CRPS pretraining:** We apply different optimisers and learning rate schedules depending on the parameter group. For the newly introduced parameters that generate AdaLN conditioning from sampled noise, we use AdamW [Loshchilov and Hutter, 2019] with a 500-step linear warmup, peaking at 2×10^{-4} before applying cosine decay down to 2×10^{-6}. Conversely, the pretrained parameters inherited from the deterministic base model are optimised using Muon [Jordan et al., 2024]. This group follows a cosine decay schedule with a longer 3000-step linear warmup and a smaller peak learning rate of 5×10^{-6}, also decaying to 2×10^{-6}.
- **Rollout Fine-tuning (RFT):** For both parameter groups, we use the Muon [Jordan et al., 2024] optimiser. We employ a cosine decay learning rate schedule decaying from 2×10^{-6} to 10^{-9} with no warmup.

B.3.5 Training Duration & Curriculum.

Deterministic model.

We train the base model for a total of 20 epochs using ERA5 reanalysis data spanning 1979–2019. With a gradient accumulation factor of 4, this results in approximately 75,000 optimisation steps.

For RFT, we restrict the dataset to the period from 2007 onwards and train for a single epoch (approx. 19,000 steps). We employ a curriculum learning strategy to stabilise training: the forecast horizon begins at 24 hours (4 autoregressive steps) and is gradually extended to 72 hours (12 steps). This extension occurs in 8 equal intervals distributed across the training duration.

Probabilistic model: Otter CRPS.

Figure 6 illustrates the training pipeline used to obtain the **Otter CRPS** and **Otter CRPS (Deep Ens)** models. It highlights the computational cost (in A100 days) and the relative performance improvements (ΔCRPS) over IFS ENS achieved after each subsequent fine-tuning stage.

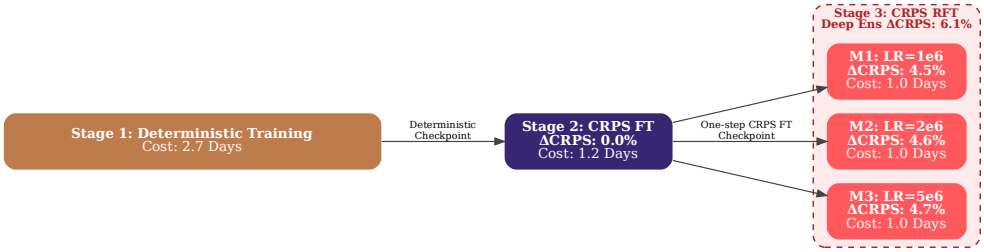


Figure 6: Overview of the training pipeline for Otter CRPS models, detailing the computational cost per stage and the progressive CRPS improvements (Δ CRPS) over IFS ENS across a 1–10 day lead time. The width of each bar is proportional to the computational time required for that stage.

Starting from the initial deterministic training phase (**Stage 1**, requiring 2.7 days), we apply a one-step CRPS fine-tuning (**Stage 2**, 1.2 days). For this stage, we fine-tune the model with the CRPS objective for 5 epochs using the same 1979–2019 ERA5 dataset used for the base model. We use an effective batch size of 16 with a gradient accumulation factor of 8.

In the final Rollout Fine-Tuning (RFT) phase (**Stage 3**), we restrict the dataset to the 2007–2019 period and apply the exact same curriculum learning strategy utilised for the deterministic model. To form a deep ensemble, the pipeline branches to train three independent models for the equivalent of 1-A100 days each, using varied learning rates (10^{-6}, 2×10^{-6}, and 5×10^{-6}). While these individual RFT models already deliver strong performance gains over IFS ENS (between 4.5% and 4.7%), aggregating their predictions as the final **Otter CRPS (Deep Ens)** provides a further substantial boost, reaching a 6.1% overall improvement in ΔCRPS.

We note that our methodology of achieving deep ensembles departs from classical deep ensembling [Lakshminarayanan et al., 2017], which typically generates members using identical hyperparameters and distinct random seeds. In contrast, we induce ensemble diversity by rollout fine-tuning each member with different hyperparameters. In this regard, our method shares closer conceptual ties with the fast ensembling approach of Garipov et al. [2018]. They construct an ensemble by employing a cyclical learning rate schedule and saving intermediate checkpoints throughout training, thereby exploring the local parameter space. We do not strictly adhere to this checkpointing procedure. Because our RFT relies on a progressive curriculum—gradually increasing the temporal unroll up to a full 3-day rollout horizon—we let ensemble member complete an entire RFT round. Consequently, rather than saving checkpoints throughout a single optimisation run, we generate the ensemble through distinct fine-tuning rounds with varied learning rates, and check that these final models do indeed achieve similar performance (i.e., between 4.5-4.7% improvement over IFS ENS). Investigating ensembling strategies that more closely adhere to the cyclical checkpointing procedure of Garipov et al. [2018] remains a promising avenue for future work.

Probabilistic model: Otter CRPS (scratch).

Figure 7 details the training pipeline for **Otter CRPS (scratch)**, highlighting both the computational cost (in A100-days) and the cumulative performance gains (ΔCRPS) over the IFS ENS baseline at each stage. In the initial phase, the model is trained from scratch with a CRPS loss for 15 epochs on the 1979–2019 ERA5 dataset, matching the deterministic model's data configuration. This stage



Figure 7: Overview of the training pipeline for Otter CRPS (scratch) models, detailing the computational cost per stage and the progressive CRPS improvements (Δ CRPS) over IFS ENS across a 1–10 day lead time. The width of each bar is proportional to the computational time required for that stage.

employs an effective batch size of 16 and a gradient accumulation factor of 8. Once the base CRPS checkpoint is obtained, the pipeline advances to the Rollout Fine-Tuning (RFT) phase, which is executed identically to the **Otter CRPS** model.

Probabilistic model: Otter CRPS with AdaLN.

The training pipeline for the **Otter** model equipped with AdaLN closely mirrors the process outlined in fig. 6. Stage 2 (CRPS fine-tuning) is executed for 5 epochs on the 1979–2019 ERA5 dataset. To obtain a single base model, we perform Rollout Fine-Tuning (RFT) for a single epoch on the 2007–2019 ERA5 data, employing the same curriculum learning strategy used for the deterministic model. To construct a deep ensemble version of the AdaLN model, we then independently train three separate models targeting different forecasting horizons: 24 hours (4 autoregressive steps), 48 hours, and 72 hours. However, because this AdaLN variant ultimately underperformed the model utilising MC Dropout (see fig. 4), we did not pursue further deep ensemble configurations akin to those used for **Otter CRPS** (i.e., utilising the same RFT horizon and varying the learning rates).

C Additional Results and Evaluation Details

C.1 Additional Evaluation Details

As mentioned in the main text, we follow the methodology in Couairon et al. 2024 and provide an aggregated skill score relative to a base model (IFS HRES for deterministic models, IFS ENS for probabilistic). This is computed as follows:

$$\text{metric-ss}(\text{model}) = \frac{1}{|\mathcal{V}|} \sum_{v} \left(1 - \frac{\text{metric}_v(\text{model})}{\text{metric}_v(\text{ref})}\right), \tag{6}$$

where $\mathcal{V}$ represents the set of key headline variables we use (Z500, Q700, T850, U850, V850, T2m, SP, U10m, and V10m), $\text{metric}_v(\text{model})$ is the metric achieved by a specific model on the variable v, and ref refers to the reference model. In the deterministic case, the metric is RMSE and the reference model is IFS HRES; in the probabilisic case, the metric is the 'fair' CRPS and the reference model is IFS ENS. For a fair comparison to some of the other AI models, we only consider the common lead times (i.e., multiples of 24h as ArchesWeatherGen uses 24h as the lead time).

C.2 Base Configuration

The **Base** configuration serves as the primary reference point for all ablation studies presented in section 4.1. This architecture was established through preliminary experimentation to provide a strong balance between predictive skill and computational efficiency. The specific hyperparameters are detailed below:

- **Embedding Dimension:** $D = 1536$;

- **Depth Profile (U-Net):** $[2, 8, 4]$ blocks per stage;
- **Attention Heads:** 16;
- **FFN Activation:** SwiGLU;
- **Positional Embeddings:** Spherical + Rotary (RoPE);
- **Regularisation:** Weight decay 0.15, Dropout 0.1, DropPath 0.1;
- **Optimiser:** Muon (momentum $\mu = 0.95$);
- **Learning Rate:** Cosine decay (max 2×10^{-4}) with a linear warmup of 500 steps;
- **Training Budget:** 20 epochs with a gradient accumulation factor of 4 (approx. $75,000$ optimisation steps).

C.3 Additional Deterministic Results

C.3.1 Additional short-horizon metrics

While fig. 3 presents ablation results averaged over the 6h to 24h window, we also report the metrics specifically at a 24-hour lead time for the various configurations. The overall trends remain consistent across both evaluation settings.

C.3.2 Longer-horizon evaluations

We further analyse the temporal evolution of RMSE skill relative to IFS HRES in fig. 8. We compare base Otter and Otter-XL against two representative deterministic baselines: GraphCast and ArchesWeather-4M. While the Otter models and GraphCast share a similar fine-tuning schedule with a rollout horizon of 3 days, ArchesWeather-4M uses a more extensive protocol extending to 4 days. In terms of computational efficiency, base Otter Weather is drastically more accessible: GraphCast requires a training budget approximately $450\times$ larger, while Arches-4M demands over $6\times$ the resources. Even our higher-performance configuration, Otter-XL, remains highly efficient—operating at just below $\approx 1\%$ of GraphCast's compute budget and requiring only 63% of the cost of Arches-4M.

We observe that all ML-based methods generally outperform the NWP baseline across the majority of variables and lead times. A notable exception is 2m temperature (T2M) at short lead times, where the higher resolution of the NWP model likely provides an advantage. Compared to GraphCast (navy), Otter Weather (coral) exhibits lower skill at short lead times, though this gap diminishes as the rollout horizon extends. Impressively, Otter-XL (dark red) significantly closes this margin: it remains highly competitive at short lead times and outperforms GraphCast at longer horizons. Crucially, this is achieved with a computational budget two orders of magnitude smaller, underscoring the efficiency of our approach.

Against ArchesWeather-4M (dark green), base Otter (coral) generally shows superior performance at short lead times, particularly on wind-related variables. Otter-XL overperforms by a wide margin up to ≈ 7 day lead time. While ArchesWeather-4M shows improved retention of skill at longer horizons—likely a direct consequence of its extended 4-day rollout fine-tuning—it is widely recognised that deterministic methods suffer from over-smoothing at these ranges. Rollout fine-tuning can only partially mitigate this intrinsic limitation; consequently, we view the exploration of probabilistic methods as the most effective strategy for long-horizon forecasting and a primary direction for future work.

C.3.3 Latitude-decomposed Effect of Attention Masking.

Section 4.1 shows that omitting the explicit attention mask has a negligible effect on global RMSE. Any penalty from information leaking across the poles would, however, only be concentrated in the polar regions, which account for a small fraction of the grid cells and could therefore be masked by a global average. Table 4 decomposes the effect by latitude band. Masking improves polar RMSE by 0.09–0.23% and degrades tropical RMSE by a similar amount; a paired block bootstrap shows that neither polar effect is statistically significant, while the global effect is significant but below 0.1%. The torus topology therefore does not cost any measurable polar skill at this resolution.

Table 3: Ablation study of architectural, regularisation, and training components. **Compute Cost** is relative to the baseline configuration (1.0×). Performance is reported as relative RMSE improvement over the base configuration (↑) at 24h lead time and does not involve RFT.

CONFIGURATION / VARIATION	COMPUTE COST VS BASE ↓	24H RMSE IMPROVEMENT ↑
Backbone Profile *(Dim D, Swin Blocks Per Stage):*		
$D = 1536, [2, 8, 4]$ (BASE)	–	–
$D = 1280, [3, 12, 8]$	1.13×	0.17%
$D = 2304, [1, 4, 1]$	1.00×	-0.05%
Regularisation *[Weight Decay (WD), Dropout, DropPath]:*		
$[0.15, 0.10, 0.10]$ (BASE)	–	–
$[0.15, 0.00, 0.10]$ (NO DROPOUT)	1.00×	-17.61%
$[0.20, 0.10, 0.10]$ (HIGHER WD)	1.00×	0.00%
$[0.10, 0.10, 0.10]$ (LOWER WD)	1.00×	0.14%
$[0.15, 0.10, 0.0]$ (NO DROPPATH)	1.09×	-0.96%
$[0.15, 0.20, 0.20]$ (HIGH REG)	0.93×	-0.51%
Window Size*:*		
$W = 15$ (BASE)	–	–
$W = 5$	1.02×	-1.03%
Patch Size*:*		
$PS = 2$ (BASE)	–	–
$PS = 1$ (OTTER-XL)	3.25×	5.28%
FFN Type*:*		
SWIGLU (BASE)	–	–
GELU	1.00×	-2.28%
Attention Masking*:*		
NO ATTENTION MASKING (BASE)	–	–
WITH MASKING	1.07×	0.17%
Positional Embeddings*:*		
SPHERICAL + ROPE (BASE)	–	–
ABSOLUTE + ROPE	1.00×	0.18%
SPHERICAL ONLY	0.99×	-1.18%
Optimisation & Dynamics*:*		
MUON (20 EPOCHS) (BASE)	–	–
MUON (30 EPOCHS)	1.50×	0.23%
ADAMW (35 EPOCHS)	1.58×	0.14%
ADAMW (23 EPOCHS)	1.00×	-0.24%
MUON (15 EPOCHS)	0.75×	-0.02%
ADAM (18 EPOCHS)	0.8×	-0.75%
History*:*		
4 STATES (BASE)	–	–
3 STATES	1.00×	-0.20%
2 STATES	1.00×	-0.65%

C.3.4 Additional metrics for the deterministic models.

Physical consistency. We evaluate Otter and the baselines on the PhysMetrics.Weather suite [Kasteleyn et al., 2026]. Several metrics are resolution-dependent, so table 5 regrids all baselines to 1.5° for a like-for-like comparison.

Conservation. At 1.5°, Otter-XL has the smallest water-mass drift of any model at both 120 h and 240 h, and base Otter the smallest at 120 h. Dry-mass drift for both is below 0.002%/day at 120–240 h, smaller than Pangu and GraphCast and comparable to Arches and HRES. Total-energy drift is smaller than Pangu and Arches, comparable to HRES, and roughly 1.5–2 × that of GraphCast.

Balance and vertical coupling. Hydrostatic balance and the lapse-rate distribution both depend on the coupling between pressure levels, so they test whether stacking levels along the channel dimension is harmful. At 1.5°, Otter-XL tracks GraphCast on hydrostatic deviation at every lead (-15.39 vs

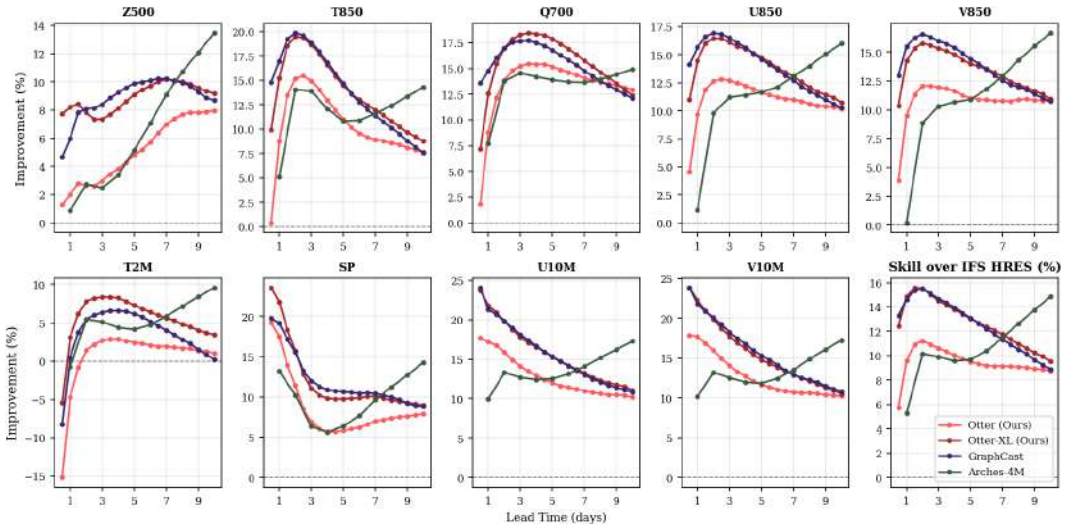


Figure 8: RMSE skill over IFS HRES comparison for lead times up to 10 days for Otter, Otter-XL, GraphCast, and ArchesWeather-4M. The last plot shows the percentage improvement over IFS HRES averaged over the headline variables.

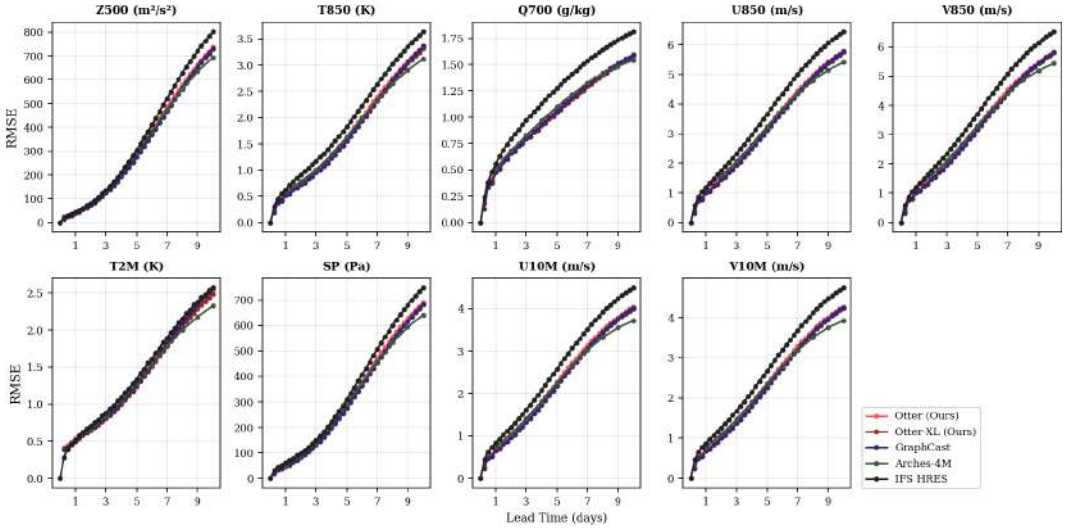


Figure 9: RMSE comparison for lead times up to 10 days. Lines represent IFS HRES (NWP baseline, black), Otter Weather (ours, coral), our more expensive Otter-XL variant (dark red), and GraphCast (navy). Otter Weather consistently outperforms the NWP baseline across most variables and lead times, remaining competitive with state-of-the-art models despite using orders of magnitude less compute. Otter-XL is competitive with GraphCast.

-15.66 at 120 h; -17.50 vs -17.94 at 240 h) and matches it on lapse-rate W_1 at 12 h (0.0169 vs 0.0168), while being worse at 120 h and better at 240 h. Base Otter has the smallest geostrophic deviation of any ML model at 240 h but the largest hydrostatic deviation at 120–240 h. We find no evidence that channel-stacking degrades vertical coupling relative to architectures with explicit level structure.

Spectra. Every model sits at the 678.5 km grid ceiling at 12 h. Otter and Otter-XL then lose effective resolution faster than GraphCast and Pangu (1019 and 1036 km vs 867 km at 120 h; 1375 and 1262 km vs 1127 km at 240 h) and are comparable to Arches, worse at 120 h and better at 240 h; spectral residual and divergence follow the same ordering. This is the oversmoothing characteristic of deterministic models trained with RMSE-type objectives, which GraphCast also exhibits, and it is the failure mode the probabilistic extension of section 4.2 is designed to address.

Table 4: **Effect of cylinder attention masking by latitude band.** Percentage change in RMSE between the non-RFT Otter base model and the corresponding model trained with the cylinder mask (positive = mask better), averaged over the nine headline variables and over 2020 initialisations. The 95% confidence interval and p-value are from a paired circular block bootstrap over initialisation times (5000 resamples) for the 6–24 h window. The polar effects are not statistically significant; the global effect is significant but below 0.1%.

	RMSE change from masking (%)			Paired bootstrap (6–24 h)	
Latitude band	6–24 h	24 h	6–72 h	95% CI	p
Tropics (30°S–30°N)	-0.09	-0.21	-0.23	$[-0.15, -0.02]$	0.009
Arctic (60–90°N)	$+0.09$	$+0.22$	$+0.17$	$[-0.06, +0.24]$	0.23
Antarctic (60–90°S)	$+0.11$	$+0.21$	$+0.23$	$[-0.04, +0.26]$	0.15
Global	$+0.09$	$+0.17$	$+0.14$	$[+0.05, +0.14]$	< 0.001

In summary, Otter is strong on conservation, comparable to GraphCast on balance at the XL scale, and weaker on spectral fidelity; it is not an outlier on physical consistency despite carrying no domain-specific inductive bias.

C.4 Additional Probabilistic Results

As mentioned in appendix C.1, for the probabilistic evaluation we use the 'fair' CRPS as the evaluation metric. When comparing to other AI models, we report performance on a 1-10 day lead time horizon, computed based on 5 ensemble members. For model selection within the Otter CRPS class, we evaluate average performance between 24–72h lead times based on 10 ensemble members. Finally, our Deep Ensemble results are derived from three distinct models, each generating three ensemble members (yielding 9 members in total). Because the fair CRPS provides an unbiased estimate of the true CRPS, the reported metrics are largely invariant to ensemble size. This is empirically supported in fig. 10, which demonstrates that the CRPS remains stable regardless of the number of members used for evaluation.

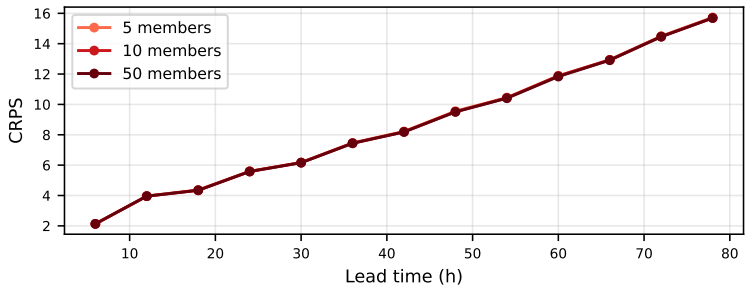


Figure 10: Evolution of the 'fair' CRPS with lead time for different number of ensemble members: [5, 10, 50]. The number of ensemble members does not affect the CRPS significantly given it is an unbiased estimator.

C.4.1 Additional results for Dropout

For the MC Dropout strategy, we study three learning rate settings (5e-6, 2e-5, 2e-4), and two different optimisers (AdamW and Muon). fig. 11 illustrates the CRPS skill improvement for 1–3 day lead times (left) alongside the Spread-Skill Ratio (right). The Muon optimiser consistently achieves superior CRPS metrics compared to AdamW; however, this performance gain comes at the expense of calibration, with Muon-trained models exhibiting higher levels of underdispersion in their ensembles.

C.4.2 Additional results for AdaLN

We evaluate the sensitivity of the AdaLN noise injection mechanism by varying the noise dimension D, learning rates, and weight decay. Aside from the noise branch, we use zero weight decay for the

Table 5: **PhysMetrics.Weather at 1.5°.** Baselines are regridded to 1.5° before computing the metrics, so the columns are like-for-like. Drift metrics are in %/day and target zero; balance metrics are deviations targeting zero; lapse-rate, spectral residual and spectral divergence are lower-is-better. [a]Arches is a 24 h-step model, so no drift line can be fitted at 12 h. [b]Arches balance and spectra values are at 24 h, not 12 h.

Metric	Lead	Otter	Otter-XL	Arches	HRES	Pangu	GC	NGCM
Conservation (drift, %/day, $\to 0$)								
Dry mass	12 h	-0.0109	-0.0104	—[a]	-0.0112	$+0.0042$	-0.0126	—
	120 h	-0.0007	-0.0018	$+0.0002$	$+0.0003$	$+0.0037$	-0.0031	—
	240 h	-0.0005	-0.0019	$+0.0006$	$+0.0004$	$+0.0023$	-0.0029	—
Water mass	12 h	-0.0514	-0.0440	—[a]	$+0.1810$	$+0.0006$	-0.2481	—
	120 h	$+0.0178$	$+0.0191$	-0.0930	-0.1207	-0.2240	$+0.1773$	—
	240 h	$+0.1285$	$+0.0698$	-0.1840	-0.0981	-0.2635	$+0.1608$	—
Total energy	12 h	-0.0126	-0.0085	—[a]	-0.0054	-0.0148	-0.0067	—
	120 h	-0.0084	-0.0078	-0.0123	-0.0109	-0.0136	-0.0045	—
	240 h	-0.0105	-0.0077	-0.0151	-0.0101	-0.0174	-0.0051	—
Balance ($\Delta \to 0$)								
Geostrophic [m/s]	12 h	-0.27	-0.27	-0.62[b]	-0.12	-0.34	-0.30	-0.27
	120 h	-0.66	-0.97	-0.85	-0.29	-0.61	-1.02	-0.57
	240 h	-0.48	-1.04	-1.34	-0.29	-0.67	-1.18	-0.62
Hydrostatic [m^2/s^2]	12 h	-4.12	-3.25	-6.99[b]	-1.03	-2.84	-2.79	-4.08
	120 h	-21.07	-15.39	-10.71	-4.60	-9.18	-15.66	-10.73
	240 h	-28.76	-17.50	-24.57	-1.85	-10.15	-17.94	-12.12
Lapse-rate W_1 [K/km] $\downarrow$	12 h	0.0186	0.0169	0.0251[b]	0.0276	0.0172	0.0168	0.0175
	120 h	0.0755	0.0714	0.0558	0.0694	0.0556	0.0655	0.0503
	240 h	0.1338	0.1149	0.1354	0.1074	0.1022	0.1180	0.0939
KE500 spectra								
Eff. resolution [km]	12 h	678.5	678.5	678.5[b]	678.5	678.5	678.5	678.5
	120 h	1018.7	1035.6	865.5	678.5	678.9	866.7	678.5
	240 h	1375.2	1262.4	2059.9	678.5	679.7	1126.5	678.5
Spectral residual $\downarrow$	12 h	0.1015	0.0918	0.220[b]	0.0550	0.0823	0.0948	0.0930
	120 h	0.7189	0.7465	0.581	0.1804	0.3036	0.5849	0.2361
	240 h	0.9948	0.9423	1.252	0.2535	0.3607	0.7747	0.3106
Spectral divergence $\downarrow$	12 h	0.1356	0.1481	0.329[b]	0.0566	0.1173	0.1261	0.1504
	120 h	1.2202	1.2670	0.922	0.1935	0.4847	1.0764	0.4426
	240 h	1.7625	1.6975	2.357	0.3626	0.5958	1.4958	0.5804

backbone and omit dropout and stochastic depth. We use the Muon optimiser for the backbone and the AdamW for the noise branch. Key findings include:

- The noise branch learning rate dictates calibration: We observe a monotonic relationship where lower learning rate for the noise branch yields better-calibrated ensembles (SSR closer to 1.0). However, CRPS improvement is non-monotonic; 5×10^{-6} leads to the best performance, with 10^{-5} being too aggressive and degrading both skill and spread.
- Skill-calibration trade-off: The best model in terms of CRPS performance is underdispersed, whereas the better calibrated model show a 0.45% reduction in CRPS improvement.
- Backbone learning rate tuning improves SSR efficiently: Increasing the backbone learning rate from 2×10^{-4} to 5×10^{-4} improves the SSR from 0.932 to 0.957 with a negligible CRPS penalty.
- Marginal impacts of weight decay and D: Higher weight decay (0.2) is strictly inferior to 0.1. Furthermore, reducing D from 128 to 64 results in a marginal CRPS loss, albeit coming at an increased computational cost and parameter count.

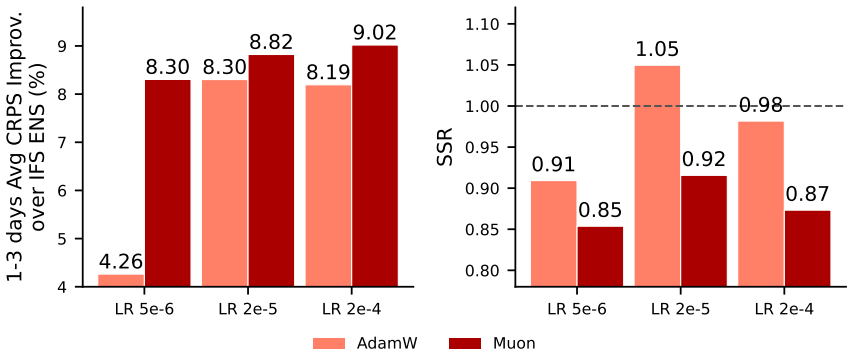


Figure 11: Skill improvement over 1–3 day lead times (left) and Spread-Skill Ratio (right) for three learning rates and two optimisers. While the Muon optimiser consistently yields superior CRPS metrics, it leads to increased underdispersion compared to AdamW-based variants.

The model we use for the AdaLN injection mechanism in the main text is $D = 128, 5\times10^{-6}/2\times10^{-4}$, weight decay of 0.1.

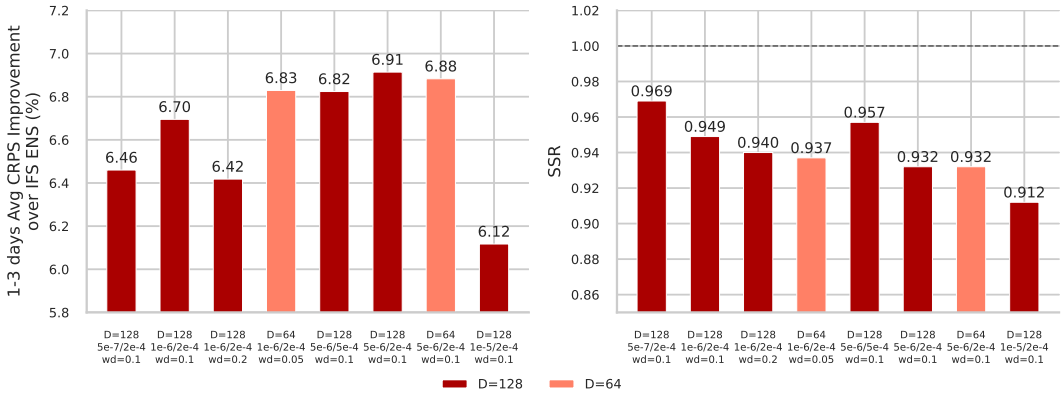


Figure 12: Skill improvement over 1–3 day lead times (left) and Spread-Skill Ratio (right) for the AdaLN noise injection strategy. The labels indicate the noise embedding dimension D, the backbone/noise branch learning rates, and the weight decay used in the noise branch.

C.4.3 From-scratch versus fine-tuning

To further evaluate the probabilistic characteristics of our proposed models, we analyse the CRPS improvement over the IFS Ensemble alongside the Spread-Skill Ratio (SSR) across lead times up to 10 days in figs. 13 and 14. These lead to the following main observations:

- **Benefits of Deep Ensembling:** Deep ensembling yields a consistent performance boost for the Otter CRPS models across all evaluated atmospheric variables. These gains are achieved at a minimal computational cost—in our case, the ensemble members were generated by varying the learning rate during a standard hyperparameter sweep for the RFT procedure (requiring only 1 A100 day per run). Because practitioners routinely conduct such hyperparameter explorations when adapting models, they naturally generate multiple distinct checkpoints. Consequently, deep ensembling can be leveraged as a highly practical byproduct of the standard development workflow, utilising these already-available models to improve probabilistic skill without incurring dedicated ensemble training overhead.
- **Calibration and Underdispersion:** While the differences in overall CRPS improvement between the fine-tuned Otter CRPS model and Otter CRPS (scratch) are relatively small, evaluating the SSR highlights distinct calibration behaviour. The fine-tuned Otter CRPS model exhibits a tendency toward underdispersion (falling below the ideal 1.0 ratio) for certain variables. In contrast, Otter CRPS (scratch) and in particular its deep ensemble

variant mitigate the underdispersion to some extent, maintaining an SSR closer to 1 for the majority of variables.

- **Comparisons to Baselines:** When benchmarking against SOTA baselines, a performance gap remains between the Otter variants and GenCast. However, Otter CRPS outperforms ArchesWeatherGen, particularly in the short-to-medium range up to approximately 6 days of lead time. We hypothesise that this specific performance profile—excelling at earlier lead times—may be a consequence of the specific RFT schedule utilised during our training process.

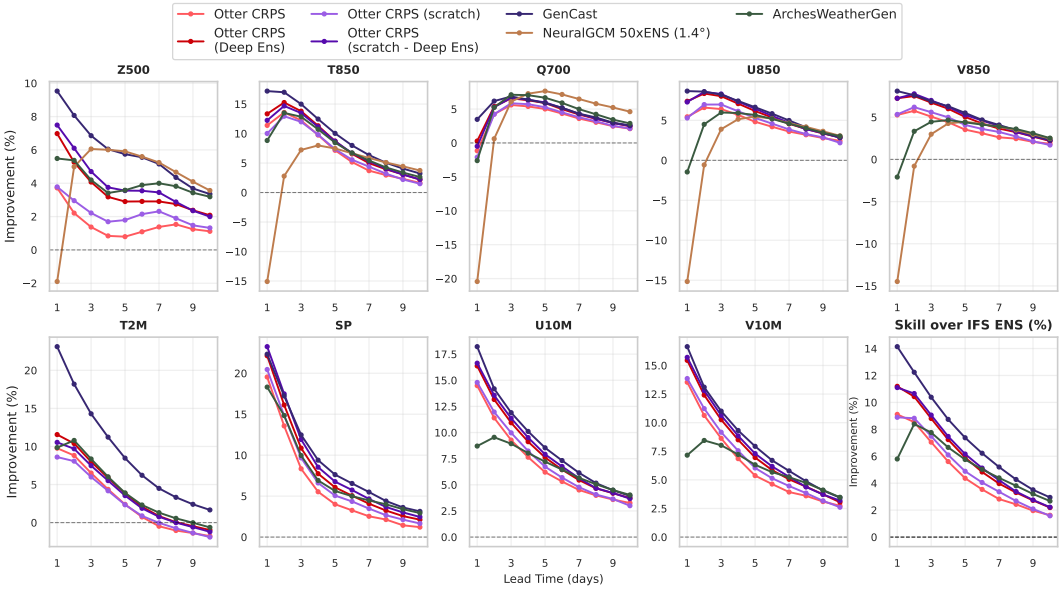


Figure 13: CRPS skill over IFS ENS comparison for lead times up to 10 days for Otter CRPS, Otter CRPS (Deep Ens), Otter CPRS (scratch), Otter CRPS (scratch - DeepEns), GenCast, and ArchesWeatherGen, and NeuralGCM. The last plot shows the percentage improvement over IFS ENS averaged over the headline variables.

C.5 Scaling the Probabilistic Model

In figs. 15 and 16 we show the CRPS improvement over IFS ENS and SSR as a function of lead time for up to 10 days, including the Otter CRPS XL variant (patch size 1). These figures confirm that the improvements from the more compute-intensive variant are not confined to short lead times but are maintained throughout the full 10-day period, with gains diminishing gradually as lead time increases.

C.6 Example Predictions

Figures 17 to 25 provide forecast visualizations for the headline variables (Z500, Q700, T850, U850, V850, T2m, SP, U10m, and V10m) across lead times of 24h, 3 days, and 10 days. We compare the deterministic Otter base model with our two probabilistic variants: Otter CRPS and Otter CRPS (scratch). For the probabilistic models, we visualize two individual ensemble members alongside the ensemble mean calculated over five samples.

C.7 Additional Results at 0.25°.

We show the per-variable results, and a comparison to FuXi in table 6. We additionally plot the Pareto Frontier evaluated at 0.25° in fig. 26.

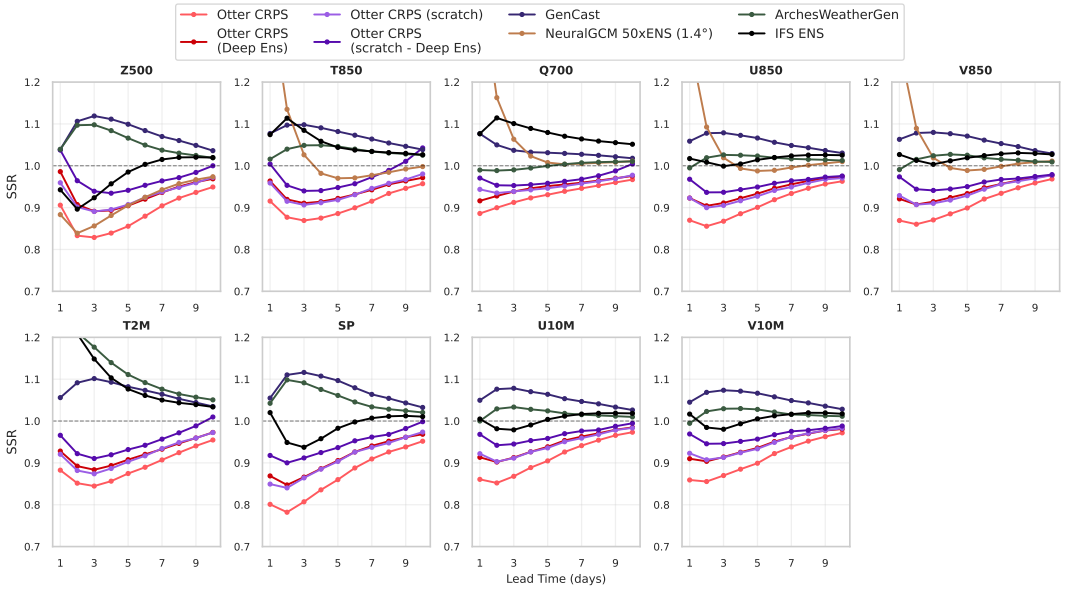


Figure 14: SSR over IFS ENS comparison for lead times up to 10 days for Otter CRPS, Otter CRPS (Deep Ens), Otter CRPS (scratch), Otter CRPS (scratch - DeepEns), GenCast, and ArchesWeatherGen, and NeuralGCM.

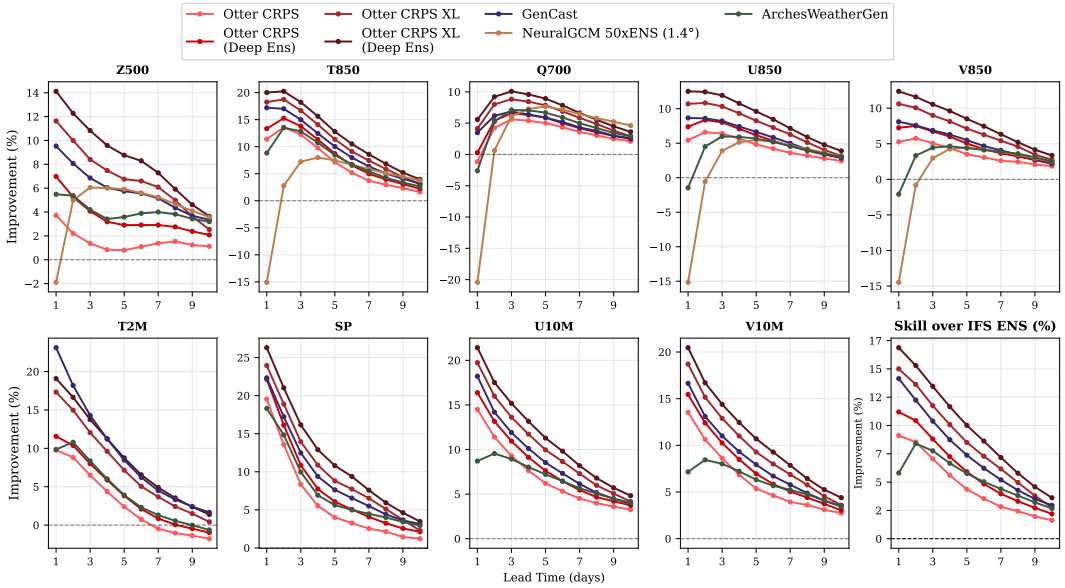


Figure 15: CRPS skill over IFS ENS comparison for lead times up to 10 days for Otter CRPS, Otter CRPS (Deep Ens), Otter CPRS XL, Otter CRPS XL (DeepEns), GenCast, and ArchesWeatherGen, and NeuralGCM. The last plot shows the percentage improvement over IFS ENS averaged over the headline variables.

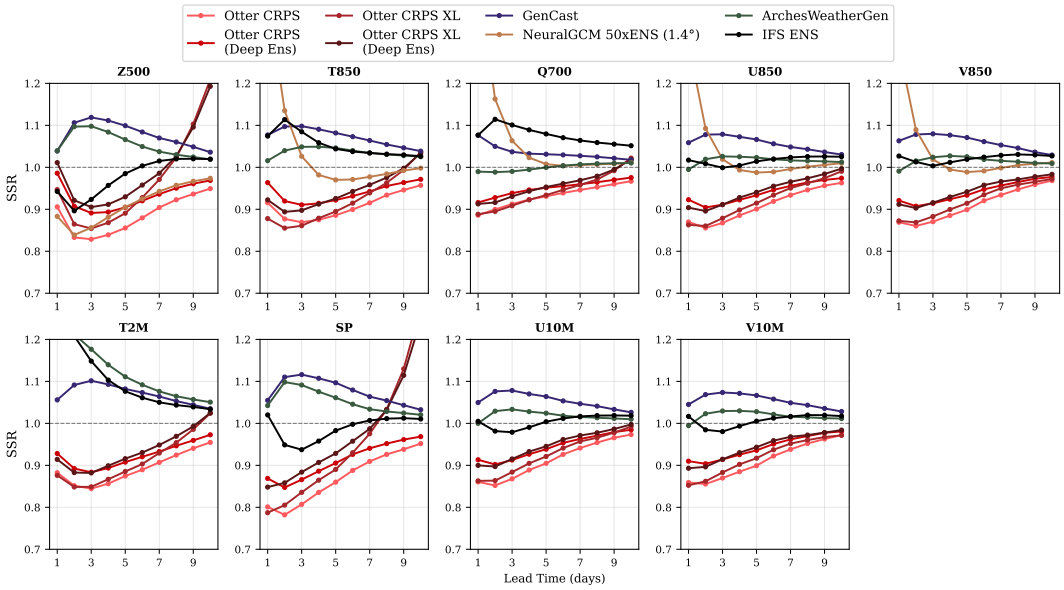


Figure 16: SSR over IFS ENS comparison for lead times up to 10 days for Otter CRPS, Otter CRPS (Deep Ens), Otter CRPS XL, Otter CRPS XL (DeepEns), GenCast, and ArchesWeatherGen, and NeuralGCM.

Table 6: **Per-variable skill at** $0.25°$**.** RMSE improvement (%) over IFS HRES, 2020. Conventions as in table 1.

Lead	Model	Z500	T850	Q700	U850	V850	T2M	SP	U10M	V10M	Mean
24 h											
	Otter-dwn	+6.3	+11.2	+19.2	+13.3	+15.6	−1.2	+18.3	+23.0	+25.4	+14.6
	Otter-0.25	+4.1	+13.4	+21.9	+19.0	+18.9	+2.5	+18.4	+28.7	+29.2	+17.3
	GraphCast	+5.7	+18.4	+24.9	+21.3	+21.2	+6.8	+19.0	+29.4	+29.9	+19.6
72 h											
	Otter-dwn	+7.6	+19.1	+23.4	+19.0	+18.4	+10.0	+11.4	+22.4	+22.5	+17.1
	Otter-0.25	+4.8	+16.9	+21.9	+17.6	+16.7	+7.0	+8.8	+20.8	+20.7	+15.0
	GraphCast	+8.5	+19.8	+23.1	+19.7	+19.1	+10.4	+12.3	+22.9	+23.1	+17.7
240 h											
	Otter-dwn	+9.3	+9.4	+15.1	+11.7	+11.8	+5.6	+9.1	+12.6	+11.9	+10.7
	Otter-0.25	+5.6	+5.8	+13.7	+8.8	+9.2	+0.6	+5.8	+9.5	+9.4	+7.6
	GraphCast	+8.7	+7.9	+14.8	+11.0	+11.5	+2.0	+8.8	+12.2	+12.1	+9.9

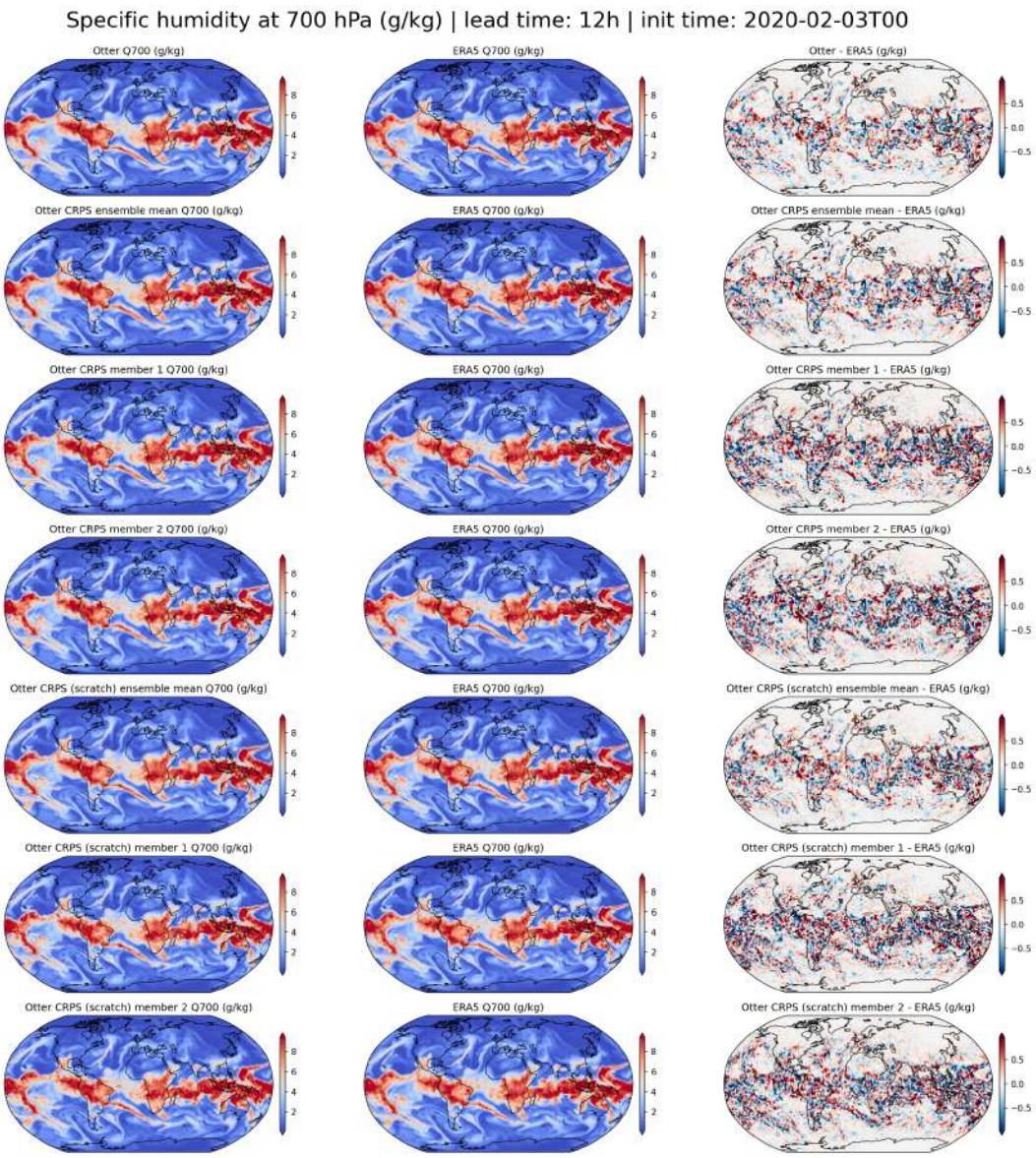


Figure 17: Visualization of Q700 at 12h lead time.

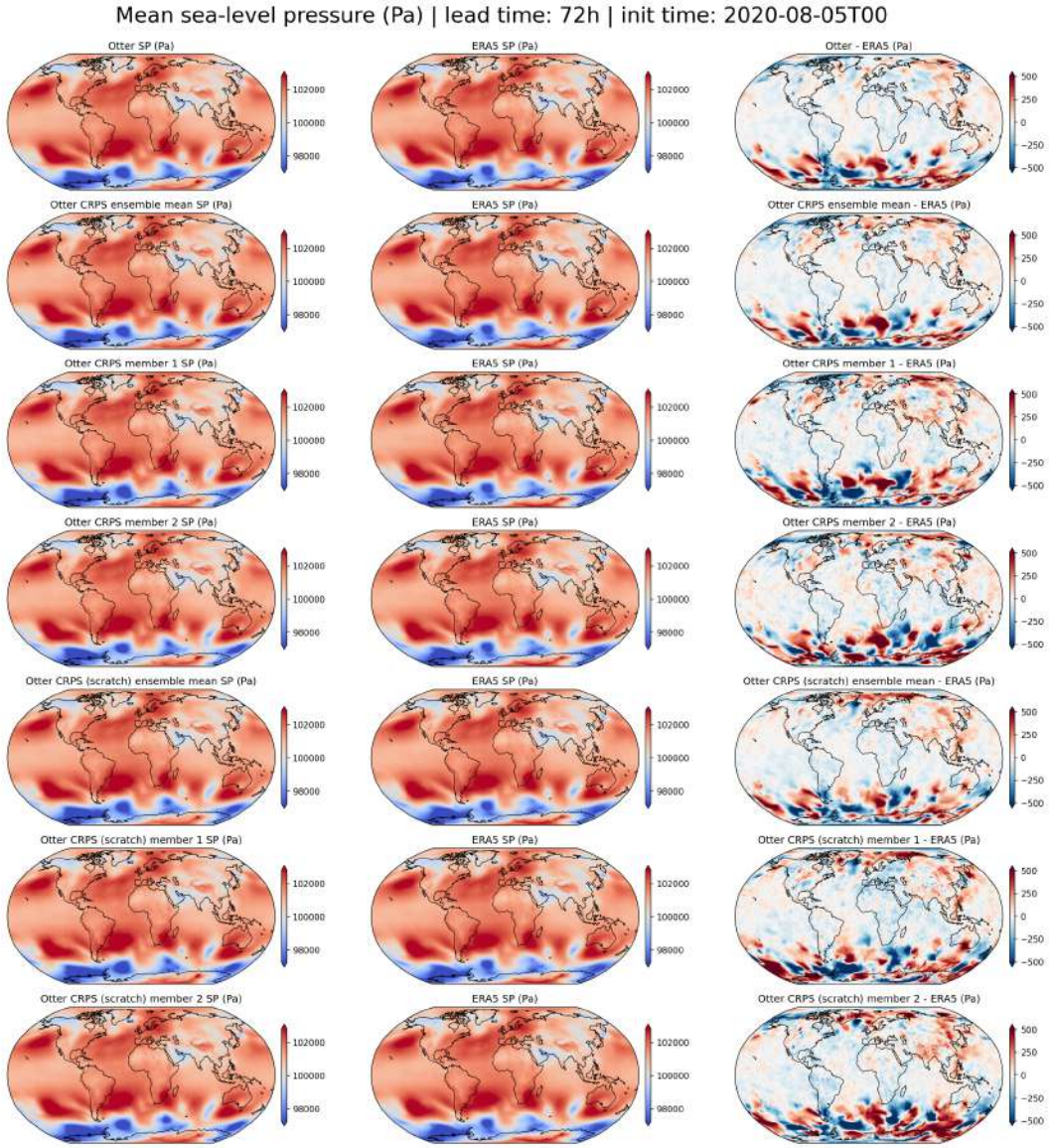


Figure 18: Visualization of SP at 3 days lead time.

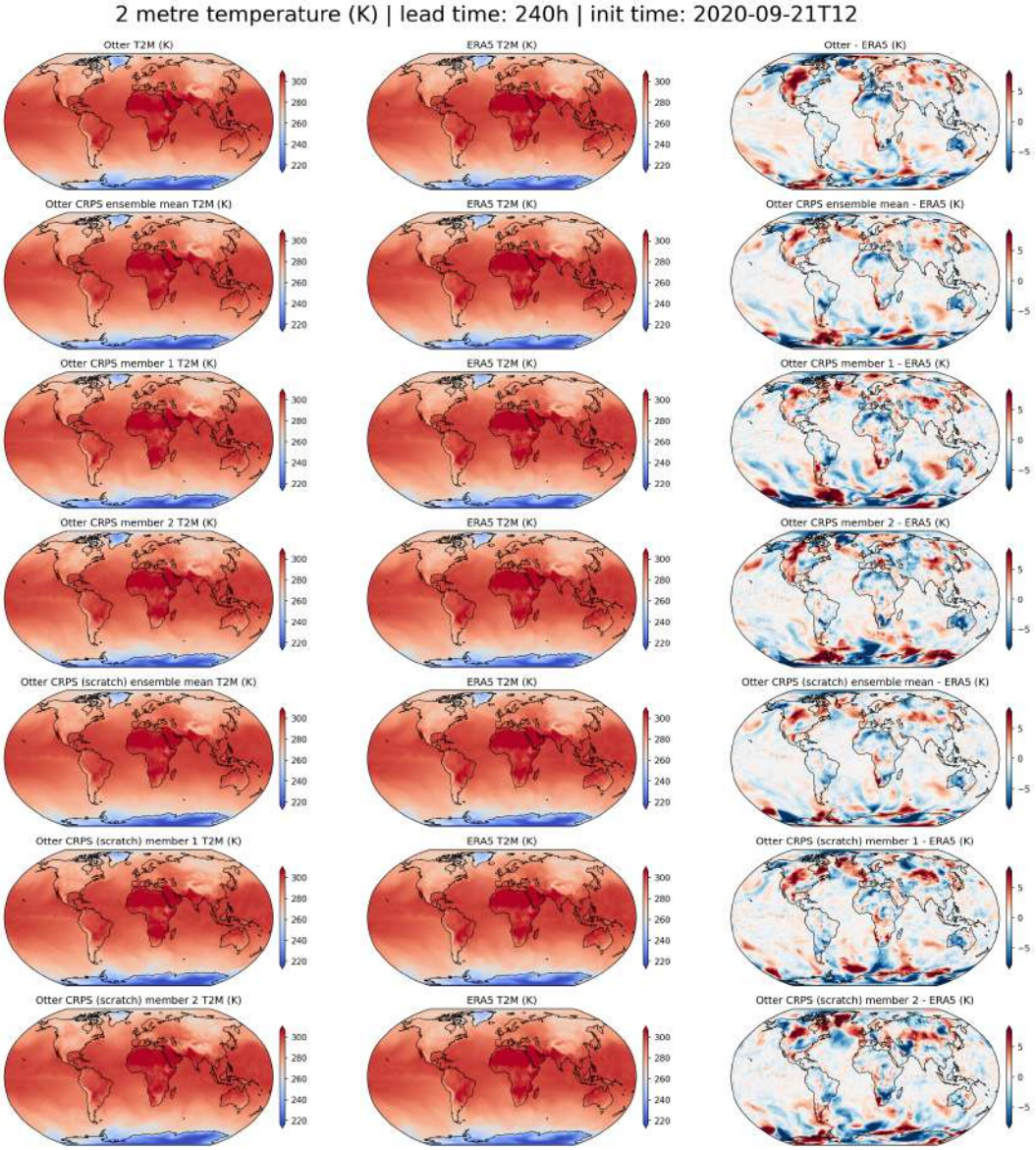


Figure 19: Visualization of T2M at 10 days lead time.

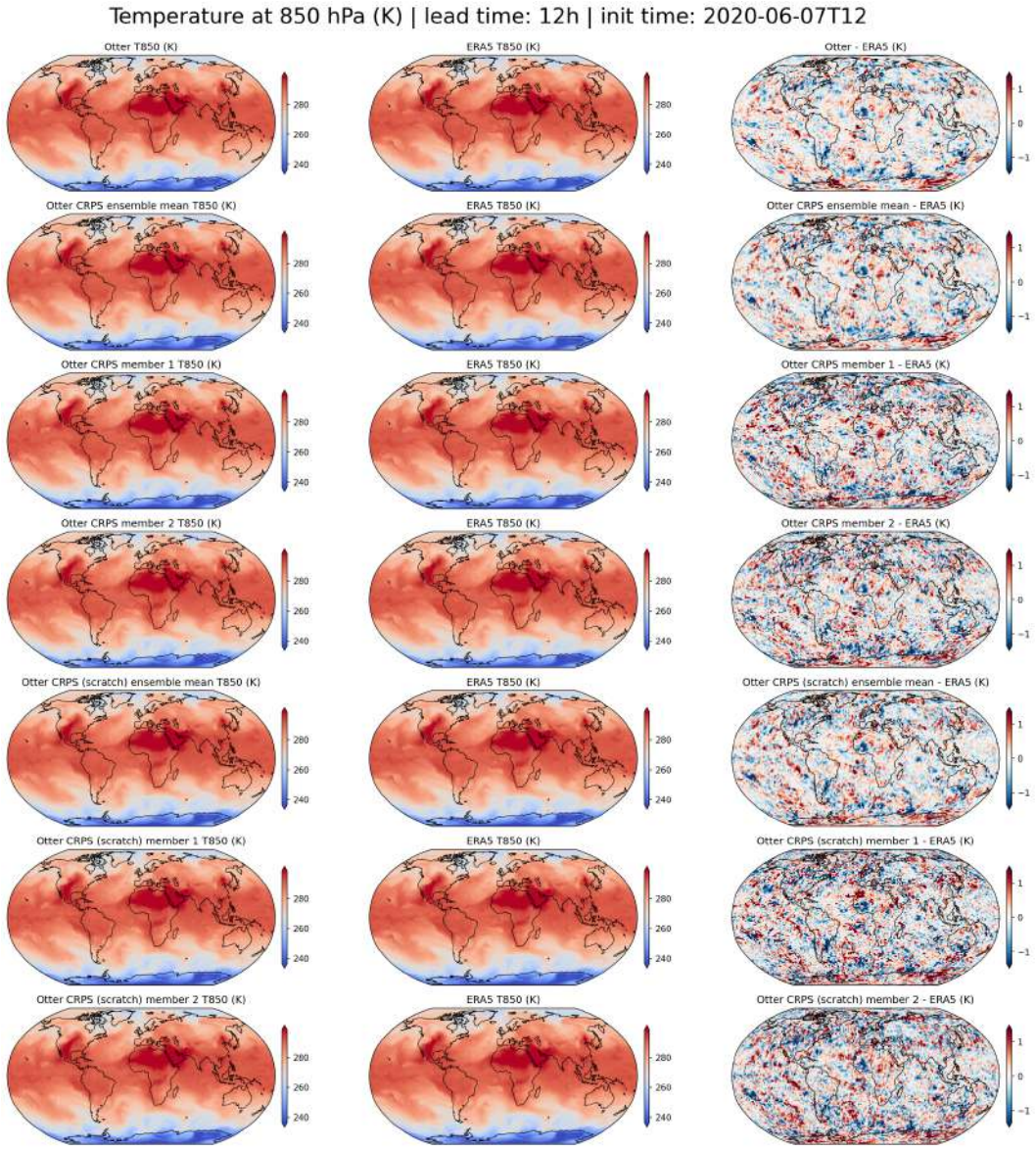


Figure 20: Visualization of T850 at 12h lead time.

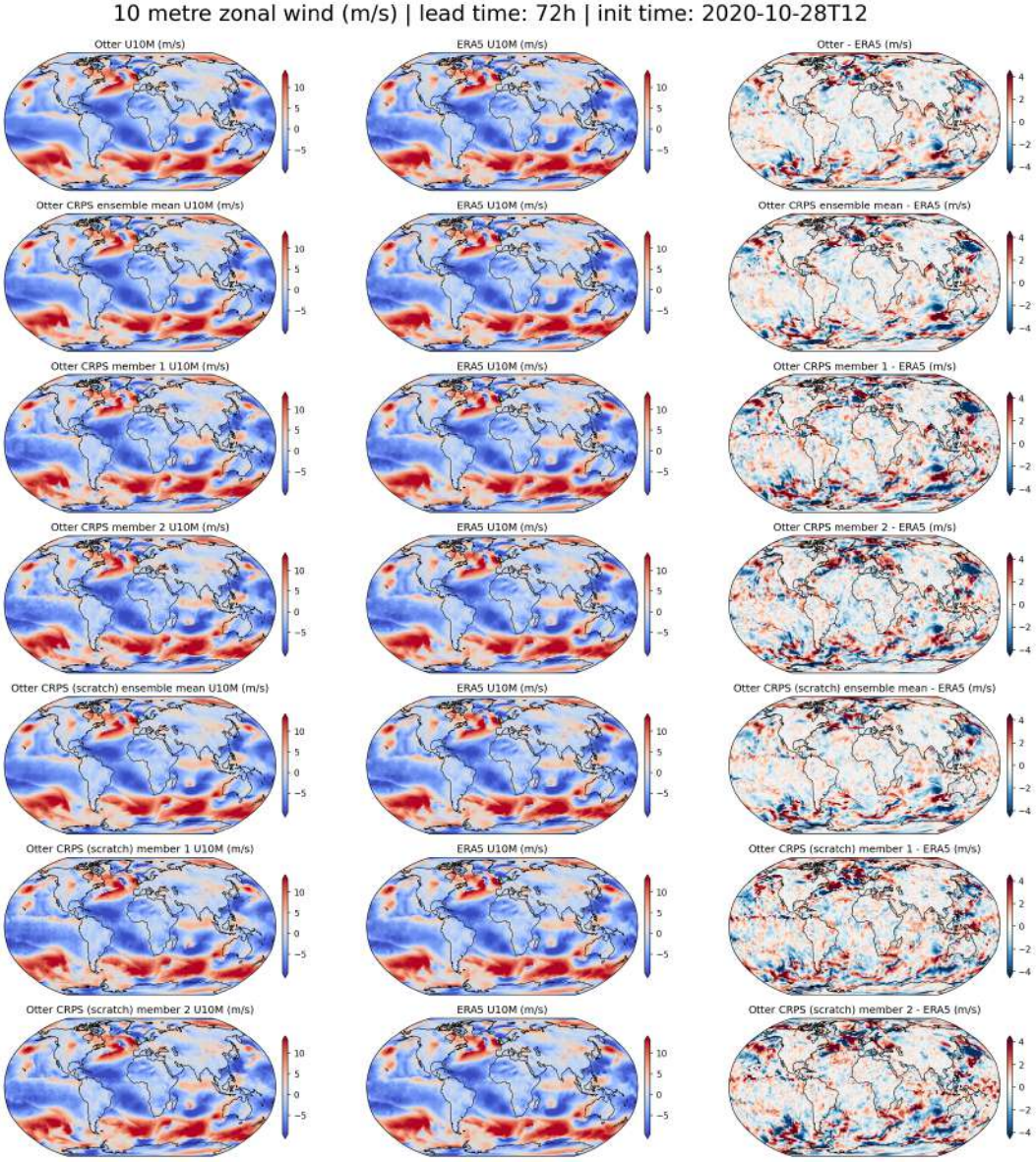


Figure 21: Visualization of U10m at 3 days lead time.

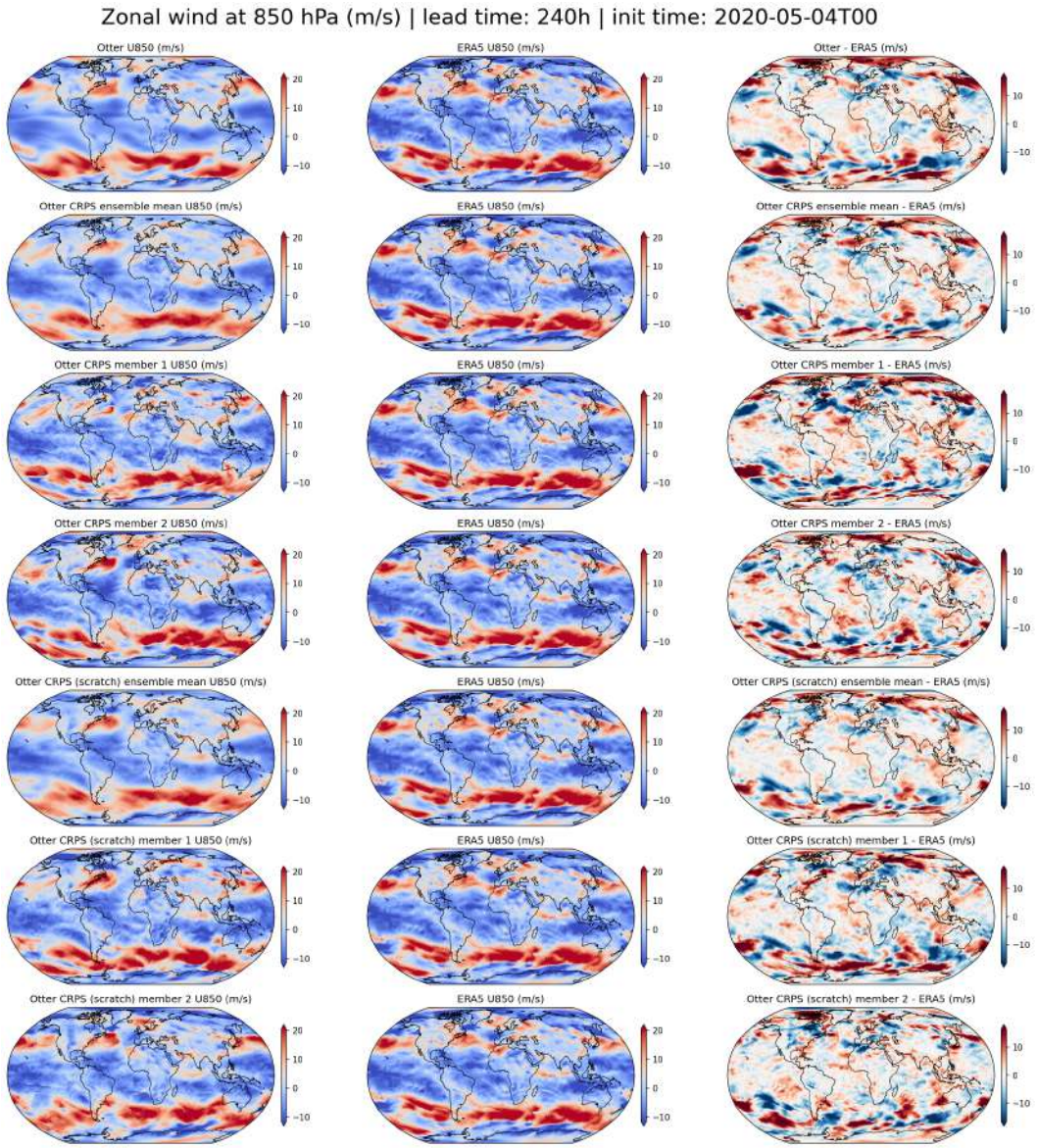


Figure 22: Visualization of U850 at 10 days lead time.

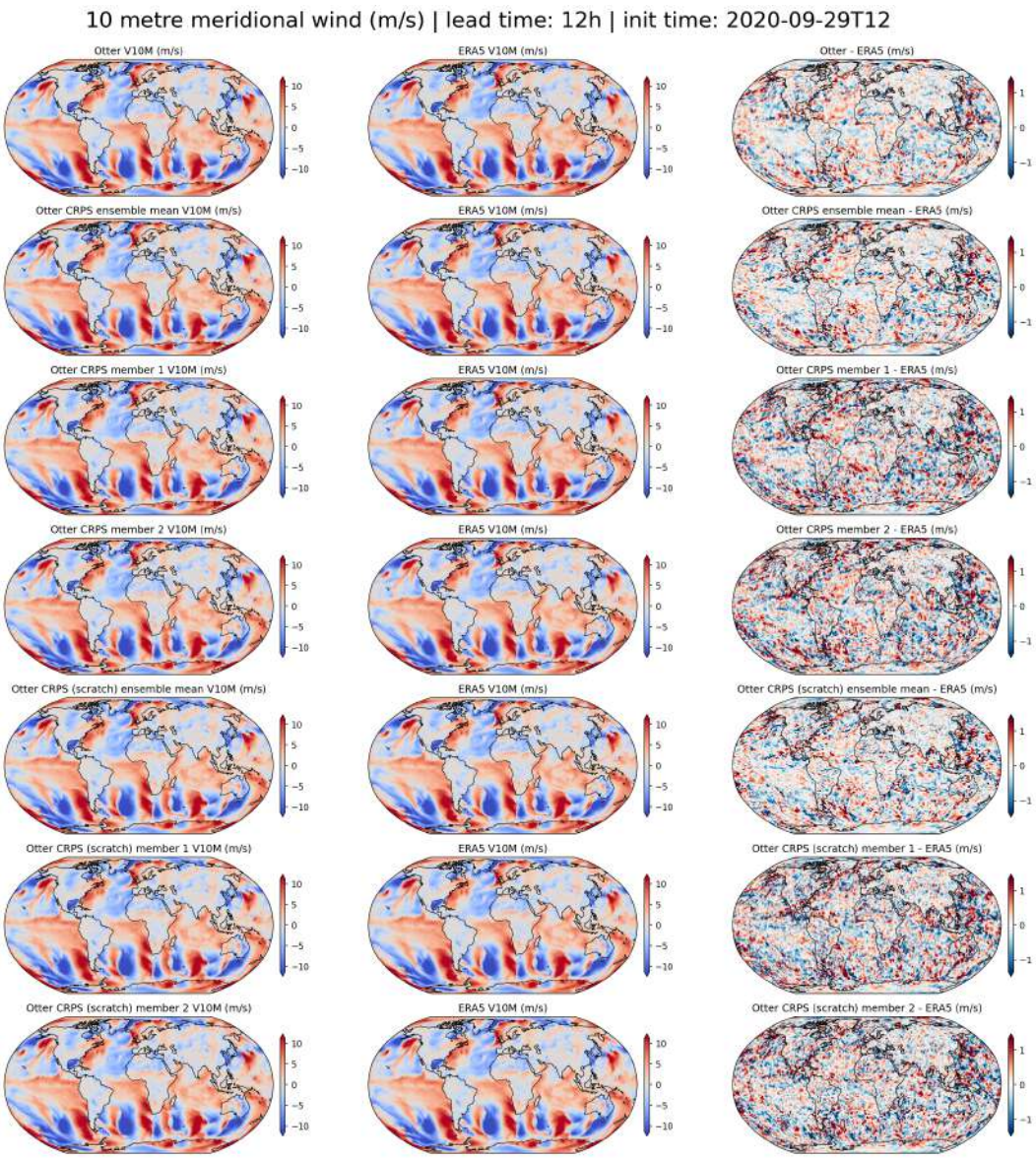


Figure 23: Visualization of V10m at 12 hours lead time.

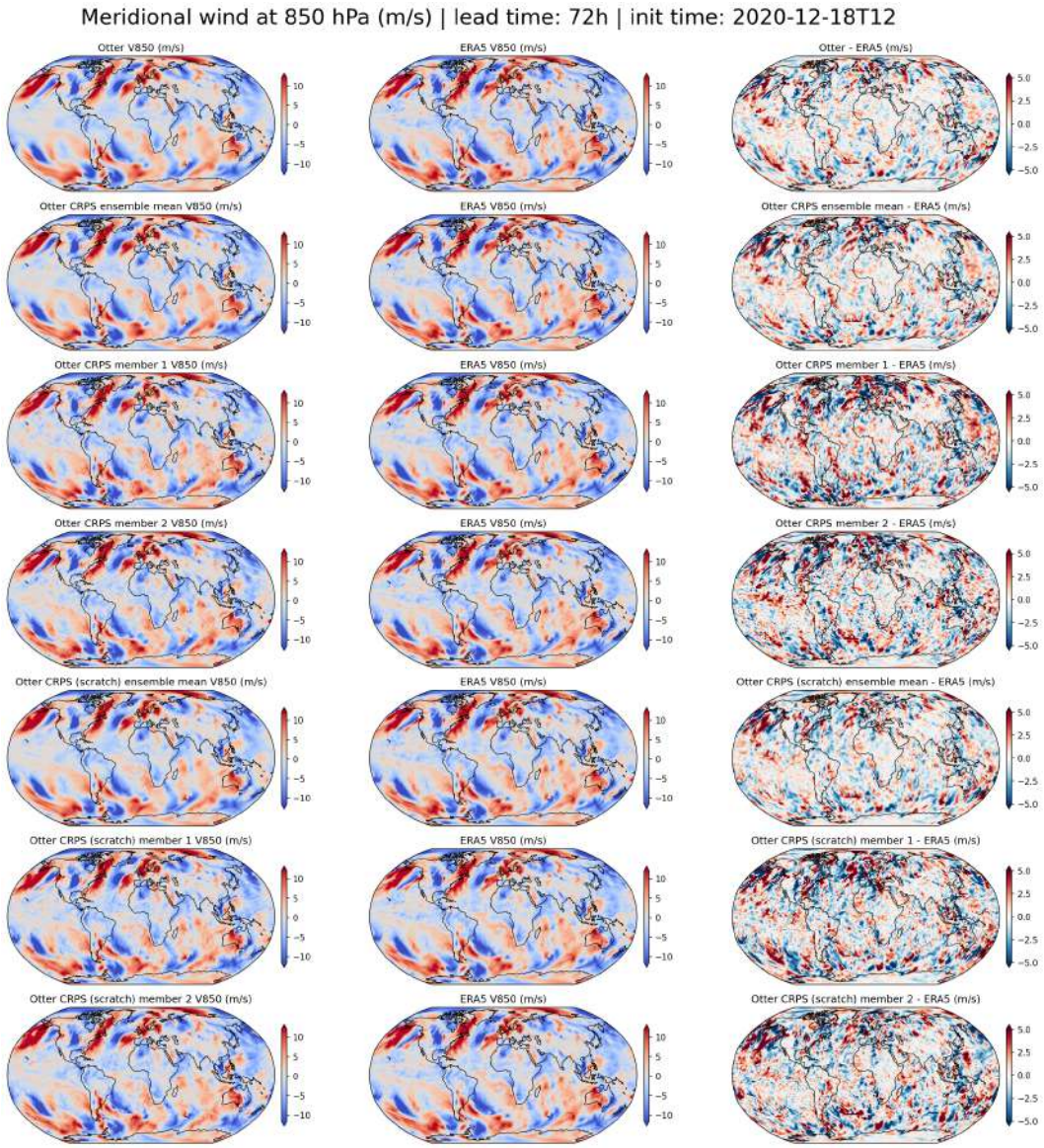


Figure 24: Visualization of V850 at 3 days lead time.

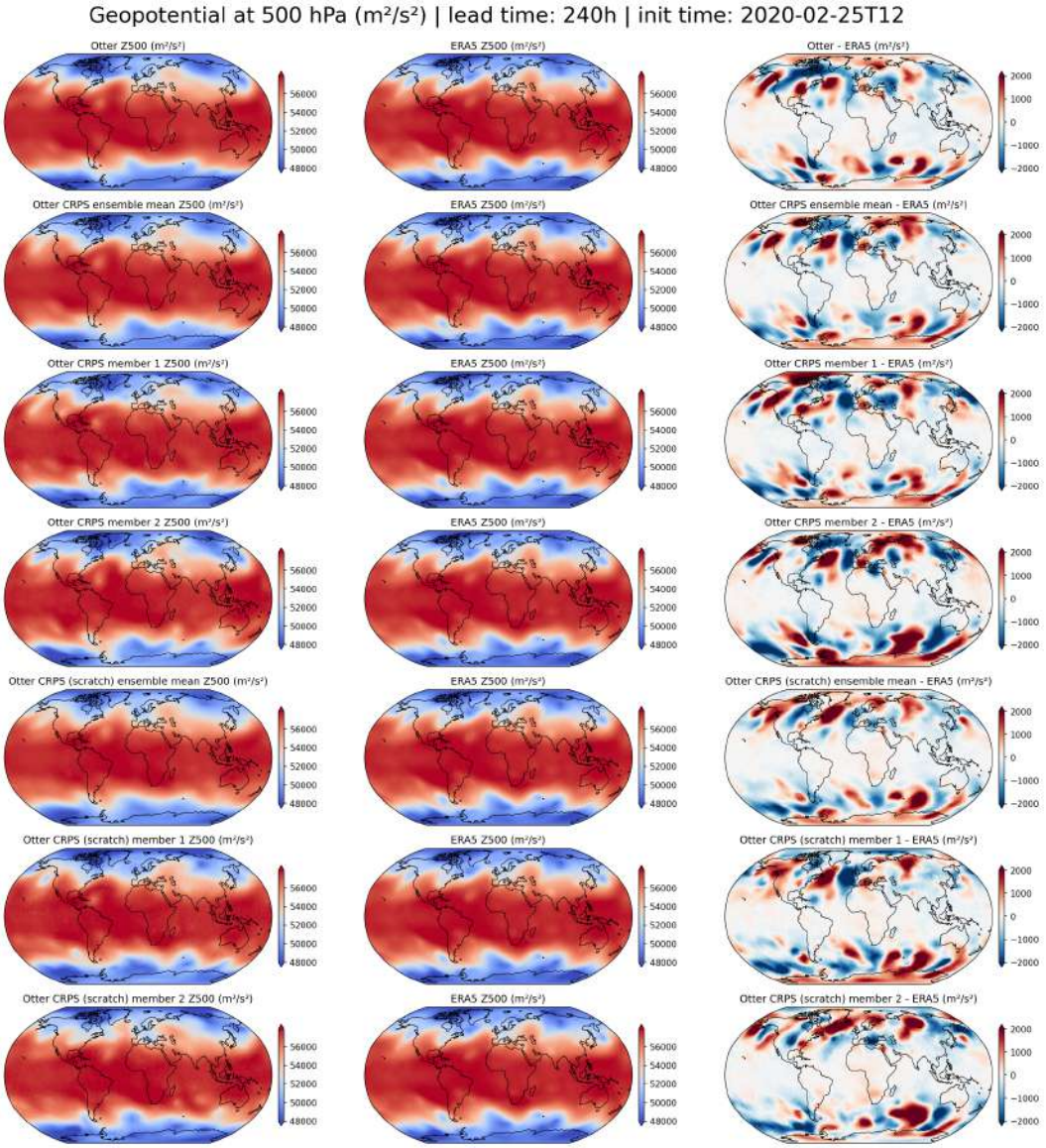


Figure 25: Visualization of Z500 at 10 days lead time.

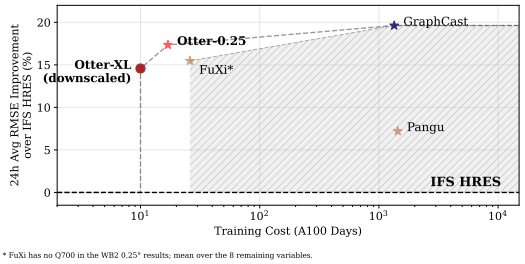


Figure 26: **RMSE Skill score at 24h over IFS HRES at 0.25° on the headline variables.** Circles (○) indicate the Otter-dwn model was built upon Otter-XL (trained at 1.5°) and post-hoc downscaled, whereas stars (⋆) indicate higher-resolution training (0.25°). The models within the Otter family maintain their computational efficiency at higher resolution.

D Cross-Domain Generality

To evaluate the generalisability of our architectural insights beyond weather forecasting, we apply the Otter base architecture to the Acoustic Scattering Inclusions dataset from *The Well* benchmark [Ohana et al., 2024]. This dataset models the propagation of acoustic pressure waves through domains with varying material inclusions—a physical regime that differs fundamentally from global atmospheric dynamics. For this evaluation, we restricted our training budget to use fewer datapoints than the budget allocated per dataset by Walrus [McCabe et al., 2026], a current SOTA foundation model for this benchmark. Despite this more limited budget, Otter demonstrates strong out-of-the-box performance, achieving a superior one-step VRMSE of 0.0030 compared to the 0.0089 reported for Walrus. These gains persist across extended rollouts (see fig. 28): in the 1–20 step horizon, Otter achieves a VRMSE of 0.020 (vs. 0.043 for Walrus), and 0.113 (vs. 0.1427) in the 21–60 step horizon. Detailed training protocols and additional metrics are provided below, and some example predictions in fig. 29.

More importantly, our ablation studies (fig. 27) suggest that the core design principles identified in the weather domain can transfer effectively to this PDE setting. The Muon optimiser once again demonstrates significant efficiency gains: at a matched 10-epoch budget, reverting to standard AdamW degrades performance by 29.6%. The architectural substitutions that proved effective for weather also appear critical here, as replacing SwiGLU with GeLU (-23.1%) or removing 2D RoPE embeddings (-5.2%) cause drops in skill. Finally, in line with our earlier findings, a balanced backbone architecture consistently outperforms both shallower, wider variants and deeper alternatives. While our current investigation is limited to a single dataset, the consistency of these findings across different physical scales is encouraging. It suggests that a well-regularised, domain-agnostic Transformer equipped with modern optimisations may provide an efficient template for modelling complex spatiotemporal dynamics. Investigating whether these design principles scale and generalise to broader PDE foundation models remains an interesting and promising direction for future work.

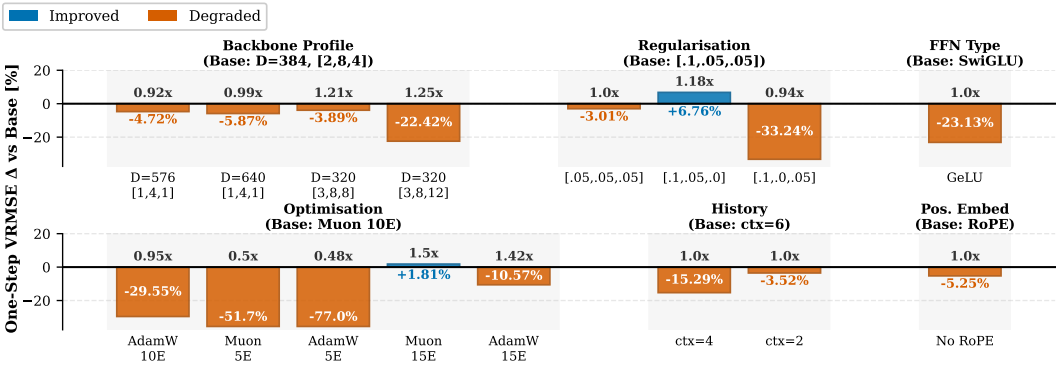


Figure 27: Ablation study on the acoustic scattering PDE task. We report the one-step VRMSE over the test set. The results confirm that findings from the weather domain (e.g., superiority of Muon, SwiGLU, RoPE, and balanced backbones) transfer to other physical systems.

D.1 Task Definition.

We provide more details on the PDE modelling task on the acoustic scattering task from The Well [Ohana et al., 2024]. This dataset consists of simulations of acoustic equations that model the propagation of pressure waves through domains with varying scattering properties (in this case, random material inclusions). The data comprises 4,000 trajectories, with each trajectory containing 101 timesteps of 256×256 resolution images. It features multiple physical channels, including scalar fields for pressure, material density, and material speed of sound, alongside a vector field representing the velocity. The density and speed of sound fields are constant, and the model only predicts the pressure and velocity channels. For additional dataset characteristics, we refer to Ohana et al. [2024].

D.2 Evaluation Details.

For evaluation we report the one-step VRMSE evaluated on the test set specified by The Well. We additionally report the rollout VRMSE in fig. 28.

Following McCabe et al. [2026] we use the following definition for the VRMSE:

$$\text{VRMSE}(u, v) = \sqrt{\frac{\langle (u - v)^2 \rangle}{\langle (u - \langle u \rangle)^2 \rangle + \epsilon}}, \tag{7}$$

where $\langle \cdot \rangle$ denotes the spatial mean operator and we add $\epsilon = 10^{-6}$ for numerical stability.

D.3 Base Deterministic Configuration.

The base configuration for PDE task is:

- **Embedding Dimension:** $D = 384$;
- **Depth Profile (U-Net):** $[2, 8, 4]$ blocks per stage;
- **Attention Heads:** 16;
- **FFN Activation:** SwiGLU;
- **Positional Embeddings:** Absolute (Spatial) + Rotary (RoPE);
- **Regularisation:** Weight decay 0.1, Dropout 0.05, DropPath 0.05;
- **Optimiser:** Muon (momentum $\mu = 0.95$);
- **Learning Rate:** Cosine decay (max 2×10^{-4}) with a linear warmup of 500 steps;
- **Training Budget:** 10 epochs with a gradient accumulation factor of 4 (approx. $84,000$ optimisation steps). The equivalent Walrus budget (in terms of data samples) is about 17 epochs.

D.4 Normalisation Strategy.

Similarly to Walrus [McCabe et al., 2026], we use the Reversible Normalisation technique. However, unlike Walrus, we rescale the backbone prediction by the per-instance RMS, rather than the RMS of the residual.

Let the input to the network $X_{ctx} = X_{t-(L-1)\times\delta:t}$. The per-instance RMS is computed over the context window:

$$\sigma(X_{ctx}) = \sqrt{\frac{1}{|\Omega|\, L} \sum_{\tau=t-(L-1)}^{t} \sum_{\mathbf{x}\in\Omega} X_\tau(\mathbf{x})^2 + \varepsilon}. \tag{8}$$

The network update with instance normalisation becomes:

$$\hat{X}_{t+\Delta t} = X_t + \text{Backbone}\left(\frac{X_{ctx}}{\sigma(X_{ctx})}, \text{Emb}(t + \Delta t) \right) \cdot \sigma(X_{ctx}), \tag{9}$$

where the backbone operates on the normalised context and its output (an increment in normalised space) is rescaled back to physical units by $\sigma(X_{ctx})$ before being added to X_t.

D.5 Training Details.

We use the mean absolute error (MAE) as the training objective. All models are trained on the NVIDIA A100-PCIe (80GB) machines for 10 epochs. This is significantly less than the equivalent budget that Walrus allocates per-dataset, which amount to about 17 epochs for the acoustic scattering dataset. Each training round takes slightly less than 2 A100 days.

D.6 Additional Results.

In addition to the one-step VRMSE presented in fig. 27, we evaluate the architectural variants using rollout VRMSE. Following the protocol of McCabe et al. [2026], these rollouts are computed over the entire trajectory length, initialised at time step 17. Consistent with our one-step findings, the majority of ablations degrade rollout performance. There are, however, some exceptions: models utilising AdamW, or a backbone configuration of $D = 640$ with a [1, 4, 1] depth profile, achieve superior

rollout metrics at a comparable computational cost. Despite these nuances, we base our primary evaluation on one-step VRMSE. Because the models are trained using a one-step objective, this metric serves as a more direct indicator of core architectural effectiveness. While long-horizon rollout stability could be addressed via techniques such as RFT, the relationship between base architecture and rollout performance remains an interesting area for future investigation.

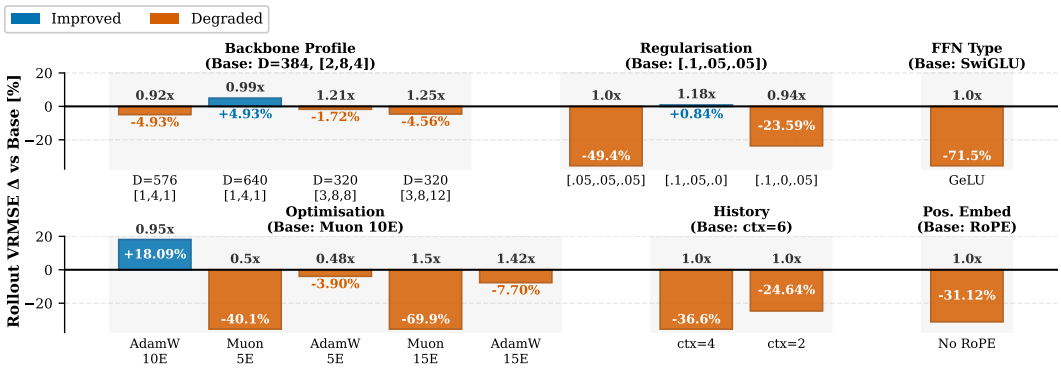


Figure 28: Ablation study on the acoustic scattering PDE task. We report the rollout VRMSE (over entire trajectory, initialised from time step 17, as performed in McCabe et al. [2026]) over the test set. The results confirm that findings from the weather domain (e.g., superiority of Muon, SwiGLU, RoPE, and balanced backbones) transfer to other physical systems.

D.7 Example Predictions.

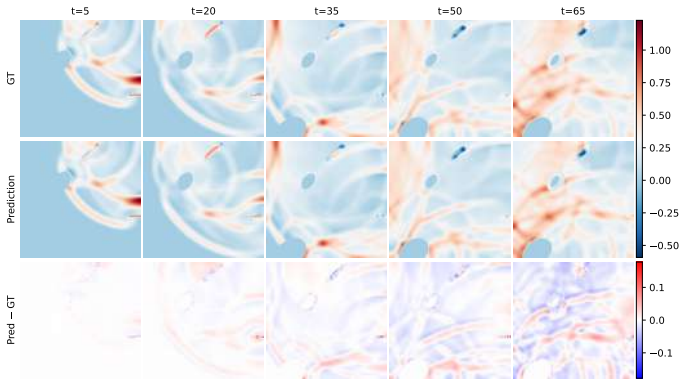
Figure 29 displays the ground truth, model predictions, and corresponding errors across five time steps of an example acoustic scattering trajectory. Results are shown for all three predicted channels: pressure and the vector velocity field.

E Limitations and Future Work

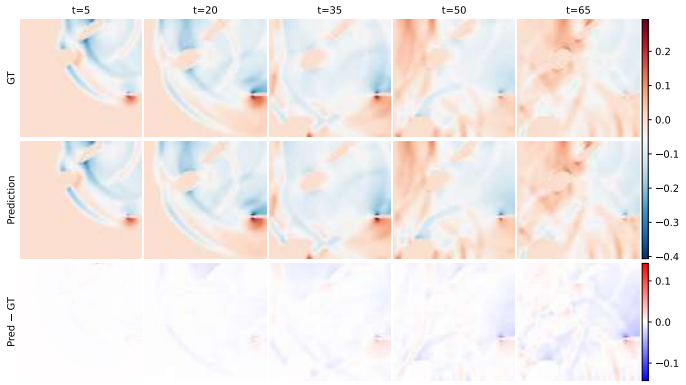
Empirical Scope & Scaling. Our findings are empirical, resulting from a budget constrained to prioritise rapid iteration. Establishing formal scaling laws or theoretical guarantees remains future work. However, Otter Weather's efficiency makes it an ideal platform for previously prohibitive scaling studies across different resolutions, downstream tasks, and architectures.

Optimal Branching for Deep Ensembles. While our results demonstrate that deep ensembling consistently yields significant improvements in probabilistic skill, the optimal methodology for constructing these ensembles remains an open question. Specifically, it is unclear at which stage of the training pipeline branching introduces the most beneficial, well-calibrated diversity. Future work should systematically investigate these trade-offs, particularly evaluating whether branching exclusively during the final RFT phase provides sufficient diversity compared to branching earlier, immediately following deterministic pre-training. Furthermore, further comparison is needed between ensembles generated via different RFT rounds on a single from-scratch model and those composed of entirely independent from-scratch training runs. A rigorous analysis of how these varying branching points impact the balance between computational cost, ensemble spread, and predictive skill will be crucial for establishing best practices in efficient probabilistic modelling.

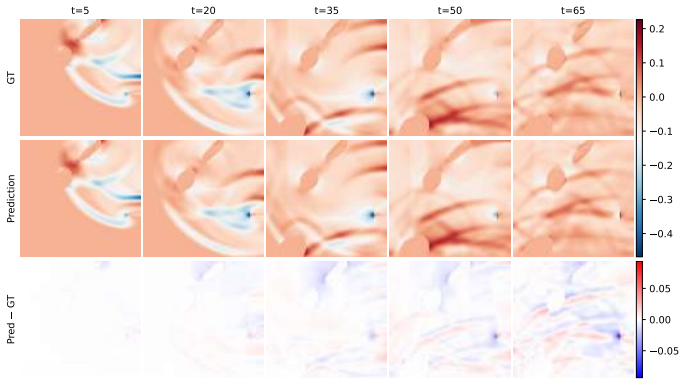
Spatial Resolution. Our ablations were conducted at $1.5°$, and only the deterministic Otter-XL has been carried to $0.25°$ (section 4.3). The natively trained Otter-0.25 uses the $1.5°$ recipe without re-tuning, so its residual gap to GraphCast at longer lead times is likely to narrow with resolution-specific tuning; we have not verified that the individual ablation conclusions of section 4.1 hold at $0.25°$, nor scaled the probabilistic model. The comparison to GraphCast also remains one-sided: we evaluate its official $0.25°$ release rather than retraining it at $1.5°$, since even at reduced resolution its cost would remain far above Otter's ($134\times$ for Otter-XL, $\sim 450\times$ for base Otter at $0.25°$), and GNN-based architectures lack the optimised hardware support that makes such retraining cheap for Transformer-based models.



(a) Pressure channel.



(b) Velocity (x-axis) channel.



(c) Velocity (y-axis) channel.

Figure 29: Example ground truth (GT) (top, predictions, and errors (prediction - GT)) for an acoustic scattering trajectory at 5 different time steps.

Inductive Biases. By prioritising general-purpose ML components, we trade some explicit physics-based inductive biases for simplicity. Our evidence that such biases are not necessary for competitive skill was obtained at $1.5°$ and has not been ablated at $0.25°$; they may matter more at higher resolution, and may help with the physical consistency of forecasts. Nor does our evidence preclude their value in general: many of our findings are orthogonal to such techniques and may compound with them, a direction we leave for future investigation.

Broader Societal Impacts. While Otter Weather significantly lowers the computational barrier to entry for global weather modelling, this democratisation carries inherent risks. By making it feasible for

practitioners without extensive meteorological or high-performance computing backgrounds to train and deploy predictive models, there is a risk of these systems being utilised in safety-critical domains (e.g., disaster management, agriculture, or aviation) without proper expert oversight. Furthermore, because we prioritise general-purpose ML components over strict physics-based inductive biases, the model may occasionally produce unphysical forecasts or miscalibrated uncertainty estimates. Over-reliance on such outputs by non-experts could lead to adverse real-world decisions. Finally, while our architecture is highly computationally efficient, the development and training of deep learning models still contribute to carbon emissions, underscoring the need to balance rapid iteration with environmental responsibility.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: figs. 1a and 1b demonstrate that Otter and Otter CRPS advance the deterministic and probabilistic skill-compute Pareto frontiers, respectively. These figures also substantiate our claims of significantly reduced training costs relative to both lightweight baselines and resource-intensive SOTA models. Furthermore, fig. 3 evidences our systematic ablation of architectural and training choices, and table 1 empirically proves that Otter reaches competitive skill at the higher 0.25° resolution.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We summarize the limitations of our work in section 5 and provide an expanded, dedicated discussion in appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper does not contain any theoretical result.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: In addition to providing the source code (https://github.com/cambridge-mlg/otter), we detail the comprehensive architectural specifications and training procedures in appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: We provide the source code alongside our submission (https://github.com/cambridge-mlg/otter). All datasets utilised in this work are open-source: the ERA5 data is publicly available via WeatherBench, and the acoustic scattering data used in appendix D is accessible through the Well dataset.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: We utilise standard train/test splits, adhering to established literature conventions for the weather forecasting evaluations and following the official dataset partitions for the PDE task in appendix D. Comprehensive training configurations, including all hyperparameters and optimiser settings, are detailed in appendix B.3 and appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

Justification: Due to the substantial computational cost inherent to training global weather models, running multiple random seeds for every configuration was computationally prohibitive, even with our efficiency improvements. To ensure a comprehensive exploration of the design space—encompassing approximately 30 independent training runs for the deterministic ablations alone—we prioritised breadth and evaluated each configuration using a single seed.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We provide full hardware specifications in appendix B.3.3, as well as make our training times clear throughout the manuscript.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research conducted conforms in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: We discuss the democratisation potential of our work, and how it could benefit the different stakeholders operating in the weather forecasting field.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: We address the broader societal implications of our research, highlighting the positive impact of democratising weather forecasting capabilities for diverse stakeholders, while also discussing the potential negative risks associated with the widespread accessibility of data-driven predictive models in appendix E.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: All external assets utilised in this work—specifically the WeatherBench and Well datasets—are properly credited via formal citations in the manuscript. All associated data and code are open-source, and our usage fully complies with their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: We include the code with the submission which is properly documented. We additionaly povide comprehensive details throughout the manuscript and the appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.