OTTER WEATHER: HIGHLY SKILLFUL MEDIUM-RANGE WEATHER FORECASTING ON A SINGLE GPU

Jonas Scholz[*1] **Cristiana Diaconu**[*1] **Aliaksandra Shysheya**[1]
Stratis Markou[1], **Richard E. Turner**[1,2]
[1]University of Cambridge, [2]Alan Turing Institute
js2731@cam.ac.uk, cdd43@cam.ac.uk

ABSTRACT

Data-driven weather forecasting models now rival or exceed the skill of traditional Numerical Weather Prediction (NWP) techniques. However, while inference remains computationally cheap, training the majority of state-of-the-art models demands a scale exclusive to well-resourced institutions, often necessitating large multi-GPU clusters and prohibitive budgets. This centralisation restricts smaller weather agencies, academic groups and startups from not only adapting models for specialised downstream tasks but also from contributing to core research on efficient forecasting methodologies. In this work, we introduce **Otter Weather**, a streamlined, deterministic forecasting model designed to democratise access to high-performance AI weather prediction. Trained on ERA5 reanalysis data and evaluated against standard WeatherBench metrics, our model achieves a 9.4% improvement over the best available deterministic NWP model at a 24h lead time. Notably, Otter is trained in three A100-days (corresponding to $70 in commercial computing costs) and is competitive with significantly more expensive SOTA methods, advancing the skill-versus-compute Pareto frontier for deterministic models. By prioritising architectural simplicity and incorporating optimised techniques from language modelling and computer vision with minimal task-specific inductive biases, Otter Weather offers a highly efficient, adaptable foundation for the broader scientific community. Finally, through comprehensive ablations of architectural, optimisation, and regularisation choices, we provide a practical recipe for efficient model training.[1]

1 INTRODUCTION

Data-driven global weather forecasting has progressed rapidly. Learned models are now competitive with—and in some metrics and regimes surpassing—strong numerical weather prediction (NWP) baselines (Lam et al., 2023; Price et al., 2024; Alet et al., 2025; Bodnar et al., 2025). This paradigm shift holds the potential to democratise weather forecasting; however, this requires sufficiently lowering the computational barriers to the field. Reducing these costs would empower a diverse range of beneficiaries—from academic institutions and under-resourced operational agencies to startups—to advance the field in two complementary ways. First, lowering the barrier to entry facilitates fundamental research into novel, efficient architectures, creating a virtuous cycle that further reduces resource requirements. Second, it enables these groups to leverage this efficiency to train specialised models—whether from scratch or by fine-tuning—for specific downstream tasks, regions, or variables. Collectively, this accelerates overall progress and expands the scope of data-driven methods, from global architectural innovations to granular, task-specific applications.

Despite this paradigm shift, the field remains far from democratised. While we are becoming less reliant on the CPU-based supercomputers required for traditional NWP, the current state-of-the-art (SOTA) data-driven methods (Lam et al., 2023; Price et al., 2024; Alet et al., 2025) have effectively 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

[*]Equal contribution.
[1]Code is provided at https://github.com/cambridge-mlg/otter

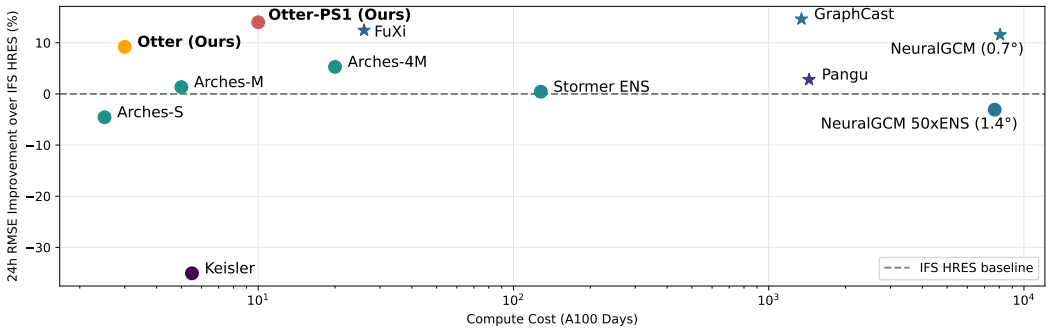

Figure 1: RMSE skill score over IFS HRES on the headline variables at 24h. Circles (○) represent models trained at lower resolutions ($1^\circ/1.4^\circ/1.5^\circ$), while stars (⋆) indicate higher-resolution training ($0.25^\circ/0.7^\circ$). **Otter Weather** and its PS1 variant advance the Pareto frontier of skill versus compute, achieving competitive skill with under 3 A100 days of compute. Baseline skill scores are computed from official WeatherBench scores. Compute costs are taken from Couairon et al. (2024)

investigate whether this paradigm can be shifted to make high-performance weather forecasting accessible to practitioners with limited resources, such as 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., 2024) 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. Consequently, we argue that such advancements should be studied within the field of weather forecasting too, and propose an ML-first solution that prioritises general-purpose, highly optimised advancements over domain-specific complexity. This approach addresses three fundamental requirements for democratisation: (1) computational efficiency, achieved by leveraging standard, optimised architectures and GPU kernels; (2) a low engineering barrier, ensuring viability on a single GPU without complex distributed systems; and (3) adaptability, by providing a general architecture that allows for easy modification for downstream tasks without navigating bespoke, Earth-specific architectural constraints. To date, ArchesWeather (Couairon et al., 2024) comes closest to satisfying these desiderata, achieving strong deterministic performance within a budget of approximately 5 A100-days.

In this work, we demonstrate that the Pareto frontier of skill versus compute can be pushed even further (Figure 1). In the low-compute regime, Otter (yellow) establishes a new standard by achieving superior skill compared to ArchesWeather (Couairon et al., 2024) in under 3 A100-days on a single GPU. Moving further up the compute spectrum, our more resource-intensive model, Otter-PS1 (red), surpasses FuXi's (Sun et al., 2024) performance at roughly $3\times$ lower compute cost and achieves skill comparable to GraphCast (Lam et al., 2023) while requiring $\approx 130\times$ less compute. Crucially, we challenge the prevailing assumption in recent literature (Lam et al., 2023; Couairon et al., 2024; Lang et al., 2024) that Earth-specific geometric priors are prerequisites for SOTA performance. While prior studies have suggested that standard vision backbones yield inferior results compared to specialised architectures, we explicitly show that this is not the case. Instead, we utilise a standard 2D Swin Transformer (Liu et al., 2021)—where variables at different pressure levels are simply concatenated along the channel dimension—augmented with modern advancements from language modelling, including Rotary Positional Embeddings (RoPE) (Su et al., 2023) and SwiGLU activations (Shazeer, 2020). To further optimise training, we employ the Muon optimiser (Jordan et al., 2024), finding it provides a superior performance-to-compute ratio compared to standard adaptive methods. Our results demonstrate that general-purpose vision backbones can define a new state-of-the-art for efficient deterministic forecasting at 1.5°. Otter Weather approaches the performance of resource-intensive models trained at higher resolutions (0.25°)—despite a $\sim 1000\times$ disparity in compute budget—showing that while complex, domain-specific inductive biases might be useful, they are not *necessary*. Our contributions are:

1. **Otter Weather.** We introduce a streamlined, ML-first 2D Swin Transformer model that establishes a new Pareto frontier for skill versus compute. Trained on a single GPU in under 3 A100-

days, it outperforms the leading low-compute baseline (ArchesWeather) and demonstrates that competitive global forecasting does not require massive distributed clusters. Furthermore, our more resource-intensive Otter-PS1 model provides another point on this frontier by surpassing FuXi at $\approx 3\times$ lower compute cost and achieving skill comparable to GraphCast with $\approx 130\times$ less compute.
2. **Reevaluating Inductive Biases.** We provide a systematic ablation study demonstrating that standard, hardware-optimised components (e.g., RoPE, SwiGLU, Muon optimisation) are sufficient for high-skill forecasting. Our results effectively challenge the necessity of the complex, Earth-specific priors prevalent in current literature.
3. **A practical training guide** We distill the key modelling and training dimensions that matter most for weather forecasting (architecture, hyperparameters, optimisation procedure, and compute), and provide a recipe to help practitioners train their own model under resource constraints. Additionally, we document engineering-heavy techniques we explored (e.g., Mixture of Experts) whose marginal gains did not justify their implementation overhead, conflicting with our goal of democratising model training.

2 Related Work

2.1 Data-Driven Global Weather Forecasting

Specialised and Foundation Models. Recent years have witnessed significant breakthroughs in data-driven weather forecasting, with learned models now matching or surpassing numerical weather prediction (NWP) baselines at a fraction of the inference cost. Models like Pangu-Weather (Bi et al., 2022) demonstrated this potential using 3D Earth-specific Transformers with hierarchical temporal aggregation strategies (i.e., training different models for variable lead times and aggregating them optimally depending on queried lead time). An alternative paradigm is that of Graph Neural Networks (GNNs) (Keisler, 2022; Lam et al., 2023; Price et al., 2024; Lang et al., 2024; Alet et al., 2025), favoured for their ability to explicitly encode spherical geometry via multi-mesh message passing. Parallel to these efforts, the FourCastNet family (Pathak et al., 2022; Bonev et al., 2025) introduced spectral approaches, utilising Adaptive or Spherical Fourier Neural Operators (AFNO/SFNO) to model global mixing in the frequency domain. However, while physically elegant, these spectral methods generally require engineering modifications to make them better suited for modern hardware (Fu et al., 2023) and complex, custom-written kernels to function efficiently (Bonev et al., 2025). Furthermore, empirical studies from partial differential equation (PDE) modelling indicate that FNOs tend to underperform U-Net-based architectures in multi-scale spatiotemporal tasks (Gupta & Brandstetter, 2023). Consequently, despite achieving SOTA performance, these architectures typically rely on heavy, domain-specific inductive biases that incur significant training overhead—often hundreds of GPU-days—and engineering complexity. More recently, the focus has shifted towards Foundation Models, such as ClimaX (Nguyen et al., 2023) and Aurora (Bodnar et al., 2025), which propose pre-training big backbones on diverse datasets to be fine-tuned for downstream tasks.

Fine-tuning efficiency. While fine-tuning specialised and foundation models on downstream tasks has proven effective (Subich, 2025; Lehmann et al., 2025), the adaptation is often hampered by challenging hardware requirements. For example, fine-tuning GraphCast for Canadian analysis data (Subich, 2025) required complex custom engineering on a 4-node A100 cluster to manage memory overhead. Similarly, adapting Aurora for hydrological variables (Lehmann et al., 2025) reportedly necessitated 32 GH200 GPUs. Such demands can be a barrier to democratisation, underscoring the need for architectures that are training-efficient by design.

Efficient Architectures and Democratisation. Recently, a counter-trend has emerged focusing on compute-efficient architectures that minimise inductive biases. Models such as Stormer (Nguyen et al., 2024) and ArchesWeather (Couairon et al., 2024) have demonstrated that competitive performance does not strictly require massive computational resources. ArchesWeather (Couairon et al., 2024), in particular, achieve strong deterministic skill with approximately 5 A100-days of training. Our work improves upon this result, surpassing Arches' performance in under 3 A100-days. Notably, while Arches relies on a specialised cross-level attention layer to reduce parameter count, we demonstrate that a standard 2D Swin Transformer—when carefully tuned—can achieve superior skill at a lower computational cost. This reduction in training cost, achieved through architectural

simplicity and careful tuning, represents a further step towards the democratisation of weather forecasting, enabling training from scratch or fine-tuning on non-specialised hardware. One of the key ingredients that helped us achieve this performance was leveraging tools and techniques that proved generally useful in ML, taking particular inspiration from the language modelling and computer vision community.

2.2 ML Breakthroughs from Vision and Language Modelling

As weather models increasingly converge with general-purpose vision and language architectures, we argue that the field requires a systematic evaluation of domain-agnostic adaptations. While works like ArchesWeather (Couairon et al., 2024) have successfully integrated specific components such as SwiGLU activations, the broader transferability of modern techniques from the LLM and computer vision communities remains under-explored and lacks rigorous ablation in the context of weather forecasting. The techniques we investigate include applying 1) Rotary Positional Embeddings (RoPE) (Su et al., 2023) instead of specialised Earth-specific embeddings, 2) the Muon optimiser (Jordan et al., 2024) to make training more efficient, 3) the importance of regularisation (i.e., dropout, weight decay, etc.), 4) Mixture of Experts (MoE) (Shazeer et al., 2017), 5) Hyper-Connections (Zhu et al., 2025), as a modern alternative to residual connections, 6) Neighborhood Attention Transformer (NATTEN) (Hassani et al., 2023), as an alternative to the Swin Transformer, and 7) Masked Autoencoder pretraining (He et al., 2021). Through ablations, we distinguish between the mechanisms that offer "out-of-the-box" gains versus those requiring involved adaptation, thereby establishing a practical recipe for efficient, domain-agnostic weather modelling.

3 Method

3.1 Problem Setup and Evaluation Protocol

Aligned with established work in data-driven weather forecasting (Couairon et al., 2024; Nguyen et al., 2024; Lam et al., 2023), our task is to predict the temporal evolution of atmospheric variables using the ERA5 reanalysis dataset (Hersbach et al., 2020). We conduct all investigations at a spatial resolution of $1.5°$. While recent state-of-the-art (SOTA) models often operate at $0.25°$ (Lam et al., 2023; Price et al., 2024; Lang et al., 2024), prior work such as Aardvark Weather (Allen et al., 2025) has established that $1.5°$ serves as a computationally efficient proxy for ablating architectural decisions, yielding principles that scale reliably to higher resolutions because the skill advantage is mainly derived from the low spatial frequency components. Furthermore, $1.5°$ remains a widely adopted evaluation standard even for high-resolution models (Rasp et al., 2023/2024), with skill at $1.5°$ being highly predictive of performance at $0.25°$. Building on this, we show that our model yields performance competitive with these higher-resolution baselines at a significantly reduced computational cost.

Our objective is to model future atmospheric states $X_{t+k\delta t}$ conditioned on a history of observations (we use 4 previous states). We employ a lead time of $\delta t = 6$ hours, where the model explicitly predicts the residual difference relative to the most recent observation rather than the absolute state. Forecasts for longer horizons are then generated via autoregressive rollout. Following Couairon et al. 2024, the state vector X_t comprises a standard set of 87 variables, consisting of $AV = 6$ atmospheric variables (temperature, geopotential, specific humidity, vertical velocity, and U/V wind components) sampled across 13 pressure levels; $SV = 4$ surface variables (2m temperature, mean sea-level pressure, and 10m U/V wind components); and five static features including the land-sea mask, surface geopotential, and sub-grid scale orography metrics (angle, anisotropy, and slope).

Following the standard WeatherBench 2 evaluation protocol (Rasp et al., 2023/2024), we utilise data from 1979 through 2019 for training and evaluate performance on the year 2020, with forecasts initialised twice daily at 00:00 and 12:00 UTC. We report performance metrics in terms of latitude-weighted Root Mean Squared Error (RMSE). To facilitate comparison between models, we also report an aggregated skill score (↑) relative to the operational deterministic NWP baseline (IFS HRES), similar to the methodology in Couairon et al. (2024) (see Appendix C.1). This aggregate metric is averaged over key headline variables: Z500, Q700, T850, U850, V850, T2m, SP, U10m, and V10m. To ensure a standardised comparison, we benchmark against established baselines rang-

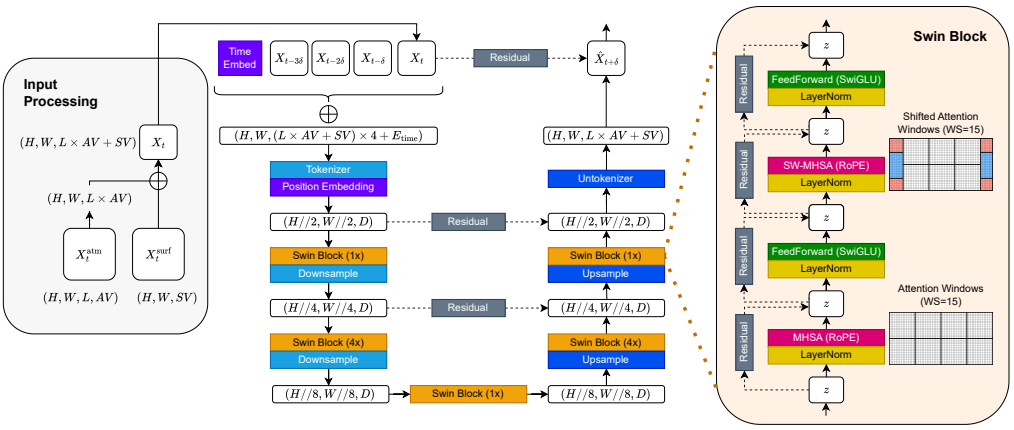

Figure 2: Otter Weather uses a simple 2D Swin Transformer, with current and previous weather variables $X_{t-3\delta} \dots X_t$ as input, and the predicted future weather state $X_{t+\delta}$ as output. Atmospheric variables at 13 pressure levels are concatenated with surface variables and time embeddings along the channel dimension. Within each Swin block, we use RoPE and by default do not apply attention masks, allowing communication between image boundaries, as shown by the shaded regions.

ing from low ($\geq 1°$) to high ($< 1°$) native resolutions, and perform all evaluations by regridding at $1.5°$. For this, we use the officially published Weatherbench results.

3.2 Model Overview: Otter Weather

Our main design philosophy is to minimise domain-specific inductive biases, favouring standard, widely supported architectural components over specialised weather adaptations. This approach prioritises hardware efficiency and ease of adaptation—allowing the model to be retrained on arbitrary combinations of atmospheric variables without requiring redesigns of variable-specific encoders or pressure-level embeddings.

Input Representation and Tokenisation. Unlike architectures that employ separate encoders for surface and upper-atmosphere variables or explicit embedding layers for pressure levels (Lam et al., 2023; Couairon et al., 2024; Bodnar et al., 2025), we simply concatenate all input variables along the channel dimension, as shown in Figure 2. This input tensor comprises SV surface variables, AV atmospheric variables, and 32 time embeddings—derived by applying 16 log-spaced Fourier features (sine and cosine) to the time variable to capture multiple temporal scales. This input tensor is tokenised using a strided convolutional layer, with patch size $P = 2$ and token dimensionality D.

Backbone: Constant-Width Swin U-Net. We employ a 2D Swin Transformer (Liu et al., 2021) arranged in a symmetric U-Net architecture comprising an Encoder, Bottleneck, and Decoder (Figure 2). We make three changes to the standard Swin architecture: first, we do not double the token dimensionality D at each stage of the U-Net, as this requires either a small initial D or excessively many parameters in the lower levels of the U-Net, neither of which we found optimal. Second, we find that using a SwiGLU feed-forward network (Shazeer, 2020) meaningfully outperforms the standard GeLU-MLP. Finally, we replace the relative position biases with 2D RoPE embeddings (Su et al., 2023), which have proven effective in language and vision modelling. Based on initial experiments, we use $[1, 4, 1]$ blocks within each U-Net stage, with 16 attention heads and $D = 2304$ as our Base configuration.

Attention Masking. Other works (Couairon et al., 2024; Bodnar et al., 2025) typically apply attention masking, approximating the topology of the Earth as a cylinder: attention is permitted across the East/West boundary, but masked across the North/South boundary. Defining and using these attention masks comes at the cost of additional implementation complexity and compute[2]. *Without* attention masks, communication is enabled across all boundaries, corresponding to the topology

[2]Many optimised attention kernels slow down substantially when supplying custom attention masks.

of a torus. We study whether attention masking is necessary, or whether the model is able to use positional embeddings to *learn* the topology instead.

Temporal and Spatial Embeddings. To capture multi-scale spatio-temporal dynamics, we employ a hybrid embedding strategy detailed in Appendix B. We follow Bodnar et al. 2025 and inject global context via absolute Fourier embeddings, encoding time and space (Latitude ϕ, Longitude λ) using continuous multi-scale Fourier features. In our ablations, we compare this against Spherical Harmonics (SH) embeddings (Rußwurm et al., 2024) for spatial features, which account for Earth's spherical topology.

Loss. The model is trained to minimise a latitude-weighted Root Mean Squared Error (RMSE) loss, a standard objective in global forecasting that accounts for projection distortion by ensuring that each pixel's loss contribution is proportional to its area. Variables are additionally weighted in proportion to their atmospheric pressure level. Following Lam et al. 2023; Couairon et al. 2024, surface-level variables are assigned weights of 1, with the exception of the 2m temperature, which is given a weight of 10.

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. We use a 500-step learning rate warmup and subsequent cosine decay. 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.

Regularisation. We use several regularisation techniques. We apply strong weight decay ($\lambda = 0.1$) as commonly seen in LLM training, and moderate dropout ($p = 0.1$) in the feed-forward layers. We also use a batch-wise variant of stochastic depth (Huang et al., 2016): with probability 0.1 (drop path probability), we skip an entire residual block for the whole mini-batch. This both reduces training FLOPs and is stable in our setting, since minibatch sizes are small.

Rollout Fine-tuning. At test time, models must provide forecasts across several lead times, a requirement typically handled through autoregressive rollouts. To explicitly train the model to produce stable rollouts and align the training objective with this test-time behaviour, we apply Rollout Fine-Tuning (RFT) after pretraining. While RFT effectively reduces error accumulation by predicting the conditional mean of the distribution at longer lead times, this tends to over-smooth forecasts, rendering them unphysical at extended horizons. Consequently, we adopt a conservative strategy: we limit RFT to a 3-day lead time—matching the protocol of GraphCast—rather than the more aggressive 4-day schedule used in ArchesWeather. Further details are provided in Appendix B.3.

4 RESULTS

We provide results for the **Base** configuration ($D = 2304$, 20 epochs), with full hyperparameters and optimisation details provided in Appendix C.2.

Efficiency vs. Performance. The current efficiency frontier for deterministic forecasting is defined by ArchesWeather (Couairon et al., 2024), which offers both a lightweight single-model version (Arches-M) trained in 5 A100-days, and a computationally heavier 4-member ensemble (Arches-Mx4). As shown in Figure 1, **Otter Weather** significantly extends this Pareto frontier. After rollout fine-tuning, our model achieves a $\mathbf{9.35\%}$ **improvement** over the operational IFS-HRES baseline (averaged across headline variables). Crucially, this performance surpasses not only the single Arches-M model (1.49% gain over HRES) but also the computationally expensive Arches-Mx4 ensemble (5.31% gain over HRES). Remarkably, this result is achieved using a single GPU with a total training budget of under 3 A100-days—more than $6\times$ less compute than the ensemble baseline.

Comparison with Large-Scale Models. Despite our significantly lower computational resources, Otter Weather remains highly competitive with industrial-scale models such as GraphCast (Lam

et al., 2023). While GraphCast was trained for approximately a month on 32 TPUv4s (a compute budget roughly $450\times$ larger), our method achieves comparable performance on headline variables at a 24h lead time. This challenges the assumption that SOTA performance requires highly specialised architectures, inductive biases, or prohibitively large compute clusters.

Long-term Rollout & Variable Analysis. Figure 3 presents the RMSE evolution for headline variables up to a 10-day lead time. Otter Weather significantly outperforms IFS-HRES across the vast majority of variables and lead times. We observe a performance gap primarily in specific surface variables at short lead times, such as 2m temperature (T2M). This is expected, as T2M dynamics are heavily influenced by fine-scale orographic features that are better resolved by the native IFS model ($0.1°$) and high-resolution baselines like GraphCast ($0.25°$). However, as the lead time increases, the influence of initial fine-scale conditions diminishes, and the performance gap between Otter Weather and GraphCast narrows significantly. We provide an additional comparison to ArchesWeather in Appendix C.3.

Scaling Compute. We additionally investigate the scaling properties of our architecture. While our primary focus is efficiency, we demonstrate that Otter Weather benefits predictably from increased resources. We trained a more expensive variant of Otter, referred to as Otter-PS1, by reducing the patch size to 1, effectively increasing the number of tokens by a factor of 4. This model was fine-tuned using a multi-step rollout schedule (detailed in Appendix B.3), resulting in a total training budget that remains accessible, at approximately 10 A100-days. This configuration significantly narrows the gap to the computationally intensive GraphCast: Otter-PS1 achieves a 14.0% gain over HRES at a 24h lead time, making it highly competitive with GraphCast (14.6%) despite requiring two orders of magnitude less compute (see Figures 1 and 3). This confirms that Otter Weather serves as both a highly efficient low-resource foundation and a scalable architecture capable of leveraging increased compute. For additional details regarding Otter-PS1, see Appendix B.3.

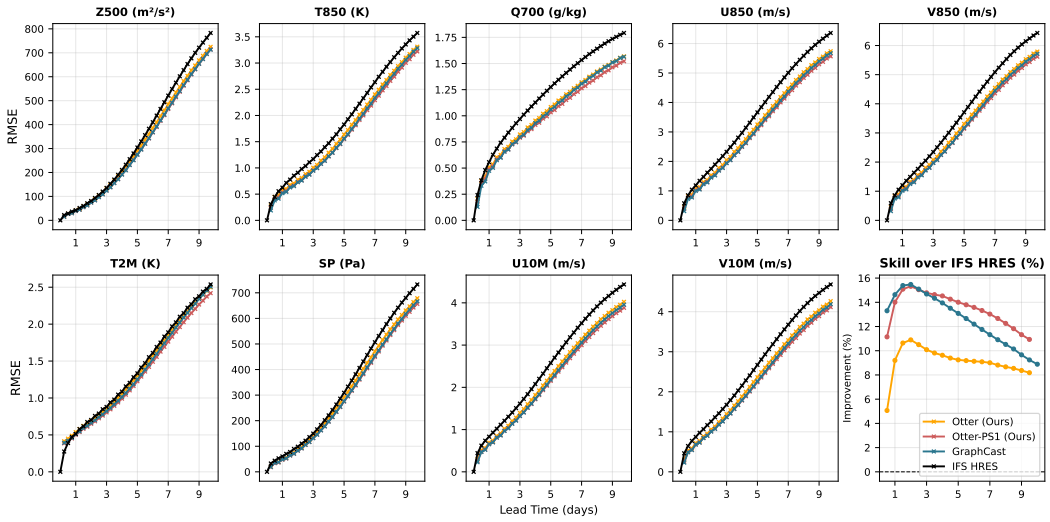

Figure 3: RMSE comparison for lead times up to 10 days. Lines represent IFS-HRES (NWP baseline, black), Otter Weather (ours, light orange), our more expensive Otter Weather PS1 variant (dark orange), and GraphCast (blue). The final subplot shows the averaged RMSE skill score across all headline variables relative to IFS-HRES. 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 PS1 is competitive with GraphCast.

4.1 Ablation Studies: Rapid Iteration

A critical advantage of Otter Weather is its rapid feedback loop. With training completing in under 3 days on a single GPU, we can iterate at a pace inaccessible to models requiring weeks of cluster time, democratising architectural search. We centre our analysis on a Base configuration (shown in Figure 2 and further detailed in Appendix C.2) established during early-stage experimentation.

Unless otherwise specified, "Otter Weather" refers to this base configuration. Using this preliminary setup, we systematically ablate key components to isolate high-impact design choices and quantify their specific computational trade-offs relative to our starting point (Table 1). As the Base configuration is not the optimal setup within every category, we expect that combining the most effective settings would lead to additional improvements, a direction we will explore in future work. Since we do not perform RFT on the ablation variants, we report the short-term forecasting performance, averaged over 6-24h lead time.

Table 1: 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 (↑) averaged over 6–24h lead time and does not involve RFT.

CONFIGURATION / VARIATION	COMPUTE COST VS BASE ↓	SKILL VS BASE (6–24H) ↑
Backbone Profile *(Dim D, Swin Blocks Per Stage):*		
$D = 2304, [1, 4, 1]$ (BASE)	–	–
$D = 1536, [2, 8, 4]$	0.97×	1.08%
$D = 1280, [3, 12, 8]$	1.11×	1.00%
$D = 2304, [1, 3, 4]$	0.95×	-0.04%
Regularisation *[Weight Decay (WD), Dropout, DropPath]:*		
$[0.10, 0.10, 0.10]$ (BASE)	–	–
$[0.10, 0.00, 0.10]$ (NO DROPOUT)	1.00×	0.83%
$[0.15, 0.10, 0.10]$ (HIGHER WD)	1.00×	0.48%
$[0.05, 0.10, 0.10]$ (LOWER WD)	1.00×	-0.41%
$[0.10, 0.10, 0.0]$ (NO DROPPATH)	1.09×	-0.15%
$[0.10, 0.20, 0.20]$ (HIGH REG)	0.93×	-1.89%
Window Size:		
$W = 15$ (BASE)	–	–
$W = 5$	1.02×	-1.04%
Patch Size:		
$PS = 2$ (BASE)	–	–
$PS = 1$ (OTTER-PS1)	3.25×	8.90%
FFN Type:		
SWIGLU (BASE)	–	–
GELU	1.01×	-1.86%
Attention Masking:		
NO ATTENTION MASKING (BASE)	–	–
WITH MASKING	1.07×	0.12%
Positional Embeddings:		
ABSOLUTE + ROPE (BASE)	–	–
SPHERICAL + ROPE	1.00×	0.13%
ABSOLUTE ONLY	0.97×	-0.55%
Optimisation & Dynamics:		
MUON (20 EPOCHS) (BASE)	–	–
MUON (30 EPOCHS)	1.50×	1.00%
ADAMW (35 EPOCHS)	1.53×	0.89%
ADAMW (23 EPOCHS)	1.00×	-0.16%
MUON (15 EPOCHS)	0.75×	-0.92%

Architectural Choices. We initially adopted a "wide but shallow" configuration with a depth profile of $[1, 4, 1]$ and embedding dimension $D = 2304$. Through ablations, we found that a more balanced architecture yields improved results. Specifically, scaling down the embedding dimension to $D = 1536$ while increasing the block count to $[2, 8, 4]$—distributing compute across all stages (initial, middle, and bottleneck)—reduced computational cost by 3% while improving skill by +1.08%. While this modification only requires a configuration change, the combinatorial search space for width/depth trade-offs is vast, making the optimal configuration hard to pinpoint without comprehensive search or scaling laws (Alabdulmohsin et al., 2024). In contrast, replacing the standard GeLU activation with SwiGLU offers a clear advantage with low selection difficulty. This simple substitution proved effective, as reverting to GeLU degraded performance significantly (-1.86%).

Finally, reducing the window size ($W = 5$) degrades performance, likely due to the constricted receptive field. Counter-intuitively, this also incurs a slight computational penalty, potentially because the overhead of managing a larger number of smaller windows negates the theoretical efficiency gains for the flash attention kernels we used.

Positional Embeddings. Regarding how to encode positional information, we find that RoPE is effective; removing it degrades performance by -0.55%. Given the availability of efficient open-source implementations, this inclusion offers a high return on engineering effort. Replacing standard Fourier features with theoretically rigorous spherical harmonics did not yield large gains (+0.13%), but they do not require additional compute, nor significant engineering effort. Surprisingly, explicit attention masking for the poles also provided negligible benefit, suggesting that the model implicitly learns the appropriate connectivity across the poles without requiring explicit constraints.

Regularisation & Dynamics. Effective regularisation proved vital; disabling dropout and increasing weight decay to 0.15 yielded the largest gains (+0.83% and +0.48%, respectively) within this category, whereas removing stochastic depth degraded performance. Aggressively scaling the regularisation also led to a decrease in skill. Regarding optimisation, on a compute-matched budget, Muon slightly outperforms AdamW (at both 20/23 and 30/35 epochs). Our initial experiments indicated that Muon converges faster, making it particularly advantageous for shorter training runs while maintaining a slight edge after 20 epochs.

4.2 Ineffective Investigations

Guided by the hypothesis that advances in LLMs and computer vision warrant investigation in weather forecasting, we evaluated several techniques prominent in these fields during the early development of Otter Weather. However, consistent with our philosophy of democratisation and simplicity, we excluded approaches where performance gains did not justify the added engineering complexity or training fragility. We document these ineffective methods to provide insight into which techniques transfer readily and which, conversely, require significant tuning or customisation.

We assessed Hyper-connections (Zhu et al., 2025) and Neighborhood Attention (NATTEN) (Hassani et al., 2023) as modern alternatives to standard residual connections and Swin blocks, respectively. Neither yielded substantial improvements over the baseline, with NATTEN introducing undesirable dependencies on custom kernels. Similarly, Masked Autoencoder (MAE) pre-training (He et al., 2021) failed to outperform standard training strategies despite the implementation overhead. Finally, while a Mixture of Experts (MoE) (Shazeer et al., 2017) configuration showed gains, the training fragility—necessitating complex load balancing and routing logic—conflicted with our goal of architectural robustness, leading us to abandon this direction.

5 Conclusion

We present Otter Weather, a highly performant model trained in under 3 A100-days on a single GPU, demonstrating that competitive global weather forecasting does not strictly require massive compute or highly specialised architectures. Otter Weather outperforms traditional NWP baselines by nearly 10% at a 24-hour lead time and competes with SOTA data-driven deterministic models. Notably, scaling compute further narrows the gap with models like GraphCast while maintaining orders-of-magnitude efficiency advantages.

This efficiency creates a transformative dual impact. On the one hand, it empowers practitioners without access to industry-scale compute to train or fine-tune models for specific, weather-related downstream applications. On the other, it reopens the field to academic and research groups, enabling them to push the Pareto frontier of weather modelling. Crucially, we observe that substantial performance improvements often emerge from the aggregation of marginal gains; thus, a rapid iteration cycle is essential, allowing researchers to navigate the design space and identify the configurations where individual improvements compound effectively. Furthermore, by scaling compute, Otter Weather approaches the performance of SOTA deterministic models such as GraphCast, while maintaining significant efficiency advantages. We discuss the empirical scope, deterministic nature, and broader limitations of our approach in Appendix D. We are excited about how this methodology impacts the broader weather forecasting community and plan to further explore the generality and scalability of our approach in future work.

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

Cristiana Diaconu is supported by the Cambridge Trust Scholarship. Jonas Scholz is supported by the Cambridge Zero/Marshall Foundation Scholarship. Richard E. Turner is supported by Google, Amazon, ARM, Improbable and the EPSRC Probabilistic AI Hub (EP/Y028783/1). Part of the computation work in this paper was funded by Microsoft AI4Good. Some of the computational experiments were generously funded by a partnership with Microsoft AI4Good.

REFERENCES

Ibrahim Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design, 2024. URL https://arxiv.org/abs/2305.13035.

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, 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*, 2025. doi: 10.1038/s41586-025-08897-0. URL https://www.nature.com/articles/s41586-025-08897-0.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast, 2022. URL https://arxiv.org/abs/2211.02556.

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*, May 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09005-y. URL https://doi.org/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.

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.

Daniel Y. Fu, Hermann Kumbong, Eric Nguyen, and Christopher Ré. FlashFFTconv: Efficient convolutions for long sequences with tensor cores. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=gPkp3n55l1.

Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized PDE modeling. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=dPSTDbGtBY.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer, 2023. URL https://arxiv.org/abs/2204.07143.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

Hans Hersbach, Bill Bell, Paul Berrisford, Gionata Biavati, András Horányi, Joaquín Muñoz Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Iryna Rozum, Dinand Schepers, Adrian Simmons, Cornel Soci, Dick Dee, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: https://doi.org/10.1002/qj.3803.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016. URL https://arxiv.org/abs/1603.09382.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, 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.

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

Dmitrii Kochkov, J. Yuval, I. Langmore, P. Norgaard, J. Smith, G. Mooers, M. K. Klöwer, et al. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024. doi: 10.1038/s41586-024-07744-y. URL https://www.nature.com/articles/s41586-024-07744-y.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Graphcast: Learning skillful medium-range global weather forecasting, 2023. URL https://arxiv.org/abs/2212.12794.

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, Matthew Chantry, Mariana Pinheiro, Peter Dueben, and Tim Palmer. Aifs-crps: Ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *npj Artificial Intelligence*, 2(1), 2026. doi: 10.1038/s44260-026-00045-z.

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, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

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, 2021. URL https://arxiv.org/abs/2103.14030.

Ankur Mahesh, William Collins, Boris Bonev, Noah Brenowitz, Yair Cohen, Joshua Elms, Peter Harrington, Karthik Kashinath, Thorsten Kurth, Joshua North, Travis OBrien, Michael Pritchard, David Pruitt, Mark Risser, Shashank Subramanian, and Jared Willard. Huge ensembles part i: Design of ensemble weather forecasts using spherical fourier neural operators, 2025. URL https://arxiv.org/abs/2408.03100.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. Climax: A foundation model for weather and climate, 2023. URL https://arxiv.org/abs/2301.10343.

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, 2024. URL https://arxiv.org/abs/2312.03876.

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. Gencast: Diffusion-based ensemble forecasting for medium-range weather, 2024. URL https://arxiv.org/abs/2312.15796.

Stephan Rasp, Stephan Hoyer, Aravind Merose, Johannes Langguth, Sebastian Deiser, et al. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *arXiv preprint arXiv:2308.15560 (Also published in Journal of Advances in Modeling Earth Systems)*, 2023/2024. doi: 10.1029/2023MS004019.

Marc Rußwurm, Konstantin Klemmer, Esther Rolf, Robin Zbinden, and Devis Tuia. Geographic location encoding with spherical harmonics and sinusoidal representation networks, 2024. URL https://arxiv.org/abs/2310.06743.

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, 2017. URL https://arxiv.org/abs/1701.06538.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

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. URL http://dx.doi.org/10.1175/AIES-D-24-0101.1.

Xiuyu Sun, Xiaohui Zhong, Xiaoze Xu, Yuanqing Huang, Hao Li, J. David Neelin, Deliang Chen, Jie Feng, Wei Han, Libo Wu, and Yuan Qi. Fuxi weather: A data-to-forecast machine learning system for global weather, 2024. URL https://arxiv.org/abs/2408.05472.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections, 2025. URL https://arxiv.org/abs/2409.19606.

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-PS1 closely approaches the performance of the best high-resolution model (GraphCast) despite using two orders of magnitude less compute.

[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.

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., 2022)	3D EARTH-SPECIFIC TRANSFORMER	0.25°	256M	1440	5.84
FUXI (SUN ET AL., 2024)	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-PS1 (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 T historical states, the input tensor to the network has dimensions (B, T, H, W, C), where B denotes the batch size, T 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 $(B, H, W, T \cdot C)$. 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:

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

We select $\lambda_{\text{min}} = 3.0$ and $\lambda_{\text{max}} = 8760$ (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 $(B, H, W, T \cdot C)$, 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 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 $\zeta_{\text{min}} = 0.1$ and $\zeta_{\text{max}} = 720$. 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:

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

Alternative: 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., 2023) 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): 2304.
- Encoder Depth: 2 Stages with block counts $[1, 4]$.
- Bottleneck Depth: 1 Block.
- Decoder Depth: 2 Stages with block counts $[4, 1]$ (symmetric to encoder).
- Attention Heads: 16 heads per block (dimension per head $d_k = 2304/16 = 144$).

- Window Size: 15×15.

This results in a symmetric depth profile of (1, 4, 1, 4, 1) 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

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 Base configuration was 46.5GB, and inference time for one autoregressive step takes approximately 0.48s.

Learning Rate Schedule. 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.

Training Duration & Curriculum. 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 for the Base configuration, and 2009 onwards for Otter-PS1, 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.

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). This is computed as follows:

$$\text{RMSE-ss}(\text{model}) = \frac{1}{|\mathcal{V}|} \sum_{v} \Big(1 - \frac{\text{RMSE}_v(\text{model})}{\text{RMSE}_v(\text{ref})}\Big), \tag{2}$$

where $\mathcal{V}$ represents the set of key headline variables we use (Z500, Q700, T850, U850, V850, T2m, SP, U10m, and V10m), $\text{RMSE}_v(\text{model})$ is the RMSE achieved by a specific model on the variable v, and ref refers to the reference IFS-HRES model.

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:** $D = 2304$;
- **Depth Profile (U-Net):** $[1, 4, 1]$ blocks per stage;
- **Attention Heads:** 16;
- **FFN Activation:** SwiGLU;
- **Positional Embeddings:** Absolute (Fourier) + Rotary (RoPE);
- **Regularisation:** Weight decay 0.1, 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 Results

We further analyse the temporal evolution of RMSE skill relative to IFS-HRES in Figure 4. We compare Otter Weather and Otter-PS1 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, 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-PS1, 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, Otter Weather (yellow) exhibits lower skill at short lead times, though this gap diminishes as the rollout horizon extends. Impressively, Otter-PS1 (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, Otter Weather (yellow) generally shows superior performance at short lead times, particularly on wind-related variables. Otter-PS1 overperforms by a wide margin up to $\approx$ 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.

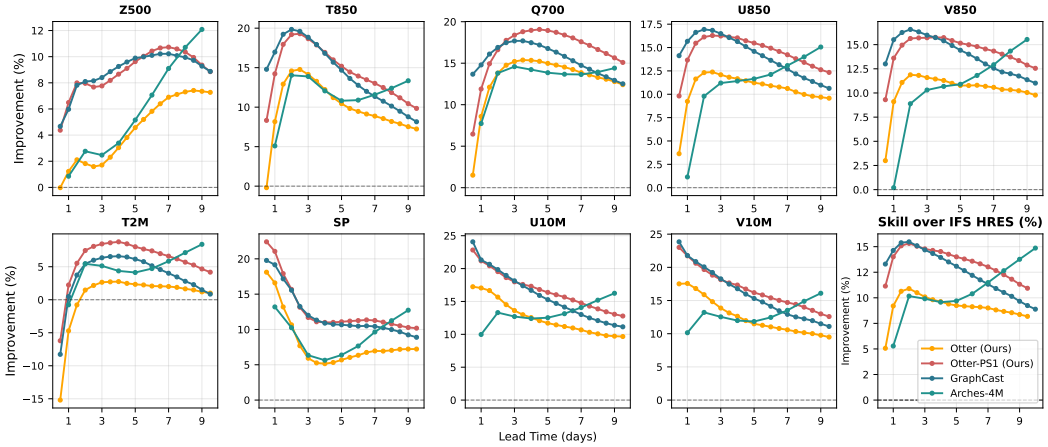

Figure 4: RMSE skill over IFS-HRES comparison for lead times up to 10 days for Otter Weather (yellow), Otter-PS1 (red), GraphCast (blue), and Arches-4M (teal).

D 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.

Deterministic Nature. We focus on deterministic forecasting, whereas the field is increasingly moving towards probabilistic approaches (Price et al., 2024; Couairon et al., 2024; Alet et al., 2025). However, since the most recent SOTA probabilistic models (Alet et al., 2025) involve adaptations of deterministic backbones, our findings are likely directly applicable to that setting.

Spatial Resolution Comparisons. We train at a 1.5° resolution, whereas some baselines like GraphCast operate at 0.25°. Because the majority of forecasting skill on test data derives from lower-frequency spatial components, this higher resolution likely contributes minimally to overall skill on the metrics we consider while significantly inflating compute constraints. While evaluating a 1.5° variant of GraphCast would provide a more direct comparison, the computational cost of the 0.25° variant is $134\times$ higher than Otter Weather's; even at a reduced resolution, its compute footprint would likely remain substantially larger. Furthermore, GNN-based architectures are notoriously difficult to implement and train from scratch, and they lack the highly optimised hardware support of Transformer-based models. Consequently, we only evaluate against the official open-source release of GraphCast operating at 0.25°.

Inductive Biases. By prioritising general-purpose ML components, we trade some explicit physics-based inductive biases for simplicity. However, this does not preclude the value of domain-specific modifications; on the contrary, many of our findings are orthogonal to such techniques, and may lead to compounding benefits—a direction we leave for future investigation.