WEATHERMESH-3: FAST AND ACCURATE OPERATIONAL GLOBAL WEATHER FORECASTING

Haoxing Du* **Lyna Kim*** **Joan Creus-Costa** **Jack Michaels** **Anuj Shetty**

Todd Hutchinson **Christopher Riedel** **John Dean**

WindBorne Systems
Palo Alto, CA 94303, USA
{haoxing, lyna}@windbornesystems.com

ABSTRACT

We present WeatherMesh-3 (WM-3), an operational transformer-based global weather forecasting system that improves the state of the art in both accuracy and computational efficiency. We introduce the following advances: 1) a latent rollout that enables arbitrary-length predictions in latent space without intermediate encoding or decoding; and 2) a modular architecture that flexibly utilizes mixed-horizon processors and encodes multiple real-time analyses to create blended initial conditions. WM-3 generates 14-day global forecasts at 0.25-degree resolution in 12 seconds on a single RTX 4090. This represents a $>$100,000-fold speedup over traditional NWP approaches while achieving superior accuracy with up to 37.7% improvement in RMSE over operational models, requiring only a single consumer-grade GPU for deployment. We aim for WM-3 to democratize weather forecasting by providing an accessible, lightweight model for operational use while pushing the performance boundaries of machine learning-based weather prediction.

1 INTRODUCTION

Numerical Weather Prediction (NWP) systems are fundamental to modern society, providing critical guidance for public safety, economic planning, and climate adaptation. However, operating NWP systems poses a high computational cost, particularly for developing regions and smaller organizations seeking to deliver essential real-time forecasting products.

Recent advances in machine learning-based weather prediction (MLWP) have demonstrated the potential to alter this paradigm (Kurth et al., 2023; Chen et al., 2023; Price et al., 2023; Lang et al., 2024a). Models such as GraphCast or Aurora have shown that neural architectures can match or exceed the accuracy of traditional NWP systems while reducing computational requirements by several orders of magnitude (Lam et al., 2023; Bodnar et al., 2024).

However, current AI approaches still face several limitations for operational use. First, despite promising results, MLWP models must continue to demonstrate improved forecast skill relative to operational NWP systems in real-time settings. Second, these models often produce excessively blurred forecasts, a direct consequence of architectural and training decisions that introduce accumulating error and can limit operational utility. Third, while less computationally demanding than traditional NWP, MLWP systems still require substantial computational resources and expertise to operate, creating barriers to widespread adoption.

We present WeatherMesh-3 (WM-3), an operational transformer-based global forecasting system that introduces advances to directly mitigate these limitations. WM-3 dramatically reduces both

*Equal contribution, corresponding authors

hardware requirements and operational complexity while improving state-of-the-art MLWP performance.

2 METHODS

2.1 MODEL ARCHITECTURE

WeatherMesh-3 takes a single state of global weather as input to predict the state of global weather any integer number of hours later. It represents the weather state on a latitude-longitude grid at 0.25 degree resolution, with surface and atmospheric variables at each grid point. Like Pangu-Weather (Bi et al., 2023), WM-3 makes heavy use of the vision transformer architecture (Dosovitskiy, 2020), and treats pressure level as a third dimension. See Appendix A.2 for details on the model architecture.

Encoder-processor-decoder architecture WM-3 consists of three high-level components: encoder, processor, and decoder. The encoder has convolution layers that map the weather state from the 0.25-degree grid to a learned, lower-resolution (2-degree) latent space. The processor then acts on the latent space any number of times, with each processor step corresponding to either one or six hours of forward time evolution. Finally, a decoder with deconvolution layers translates the latent space back into a 3D grid in physical space. A schematic of the model architecture is shown in Figure 1.

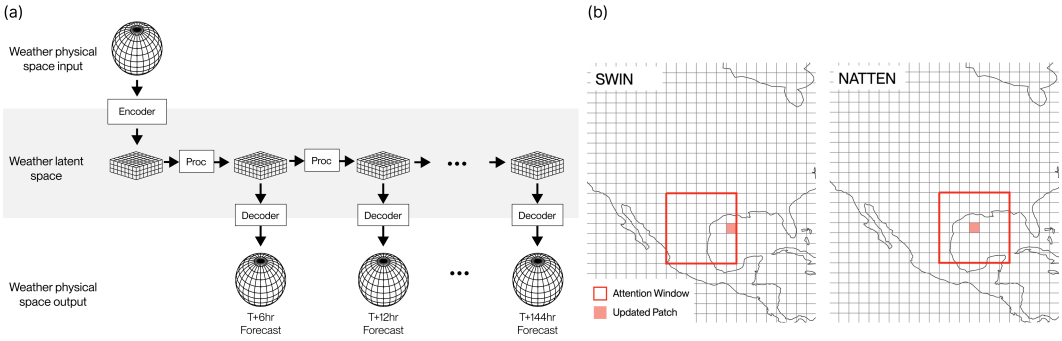

Figure 1: (a) A schematic of the encoder-processor-decoder architecture. (b) Illustration of the difference in attention window location between SWIN and NATTEN.

WM-3 uses a *latent rollout* to produce a forecast for an arbitrary integer number of hours of forecast lead time in a single forward pass. It achieves this by invoking a combination of the six-hour and one-hour processors repeatedly until the target lead time, using a simple greedy schedule. This avoids the typical MLWP rollout method of encoding a weather state, rolling forward six hours, decoding to the physical space, then re-encoding to take another six hour step forward. Keeping the atmospheric state in latent space between rollouts has three main advantages: 1) avoids extraneous encoding and decoding compute or error accumulation processes; 2) enables mixed-horizon prediction tasks in training (see Section 2.2); 3) allows modularity in every model component by working with a shared latent space.

NATTEN The backbone of the WM-3 processors is a series of neighborhood attention-based transformer (NATTEN) blocks (Hassani et al., 2023). In earlier versions of WeatherMesh, we experimented with the SWIN architecture (Liu et al., 2021), and found that NATTEN significantly improves WM-3 performance. The neighborhood attention mechanism provides a better inductive bias to the model for learning atmospheric physics due to its consistent locality of attention for information transfer between patches (Figure 1b). In addition, thanks to the performance of the NATTEN library with Fused Neighborhood Attention kernels (Hassani et al., 2024), NATTEN is faster and has a lower memory footprint than our prior SWIN implementations.

2.2 TRAINING

Pre-training WeatherMesh-3 undergoes a two-stage training procedure: pretraining on ERA-5 reanalysis data, then fine-tuning on analysis data to create an operational version for real-time use.

WM-3 is pretrained on ERA-5, ECMWF's reanalysis dataset, from 1979 to 2022, with 2020 left out of the training data for testing per Rasp et al. (2024). The six-hour processor is trained first to predict the state 6 and 12 hours later. The target timestep is then lengthened up to 120 hours (5 days, or 20 calls to the six-hour processor) according to a schedule. Once the six-hour processor is trained, we freeze the weights in the encoder and decoder and train a one-hour processor. We use Distributed Shampoo (Shi et al., 2023) for training on multiple GPUs. See Appendix A.3 for details on the training setup and A.5 for compute optimizations.

Operationalization In order to make WM-3 suitable for operational use, we replace the pretrained encoder with two new encoders that ingest analyses from European Centre for Medium-Range Weather Forecasts (ECMWF)'s Integrated Forecasting System (IFS) and the National Oceanic and Atmospheric Administration (NOAA)'s Global Forecast System (GFS). See Figure 4 for details. WeatherMesh has been running operationally since March 2024.

3 RESULTS

Accuracy Operational WM-3 achieves high accuracy relative to current operational models. To evaluate WM-3, we calculate latitude-weighted RMSE scores for our operational forecasts against ERA-5 and compare with IFS HRES over our evaluation period in 2024 (see Appendix A.4). In Figure 2, we present the evaluation scorecards of WM-3. WM-3 outperforms HRES on *all but one* of 690 targets evaluated (geopotential at 50 hPa at 10 days), with a notable 37.7% improvement over HRES on 2-meter temperature at 1 day lead time.

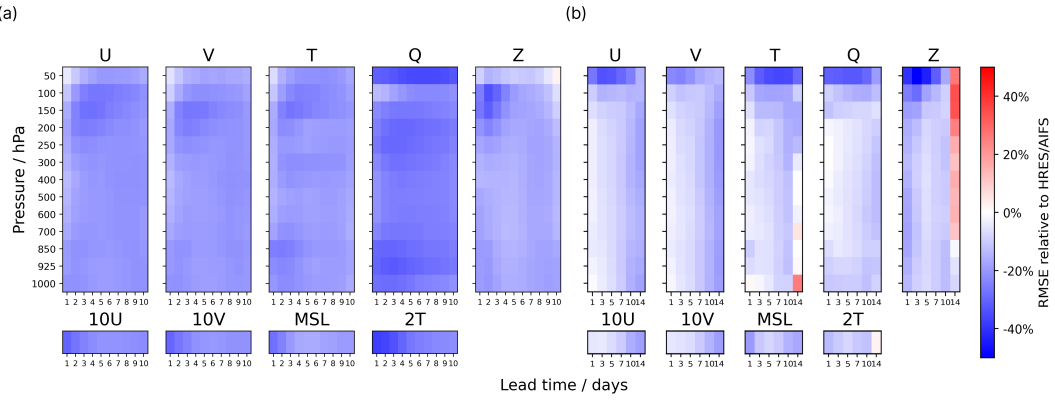

Figure 2: (a) Scorecard comparing WM-3 vs. IFS HRES across all pressure levels and surface variables. (b) Scorecard comparing WM-3 vs. AIFS.

MLWP models outperforming HRES on RMSE should be evaluated with the perspective that HRES has not been optimized for RMSE as a metric. While it is common to compute RMSE for HRES against its own analysis in literature (Lam et al., 2023; Bodnar et al., 2024), we report RMSE against ERA-5 for both models and treat ERA-5 as the best representation of ground truth.

We also report WM-3's performance against AIFS, ECMWF's own operational AI model (Lang et al., 2024a), in Figure 2(b). We note that WM-3 consistently outperforms AIFS at 50 hPa but is relatively weaker at 14 days forecast lead time, especially on geopotential (Z).

Blurring MLWP models are known to suffer from excessive blurring (Lam et al., 2023; Rasp et al., 2024; Lang et al., 2024b), where models learn to blur excessively to achieve lower training loss (usually mean squared error, MSE). In operational use, however, a blurry forecast with a lower

RMSE score is often less useful than a less blurry forecast with a slightly higher RMSE score. To evaluate WM-3 in the trade-off between accuracy and blur, we calculate a "blur score" based on spectral power at 500km wavelength (defined in Appendix A.4), and plot it against RMSE, shown in Figure 3(b). A useful reference is ECMWF's ensemble forecasts (ENS) (ECMWF, 2024), which leverage an ensemble of 51 members to achieve higher accuracy but inevitably introduces more blurring. WM-3 simultaneously achieves lower RMSE score *and* lower blur score than both ENS and AIFS on some variables and forecast lead times. Furthermore, for most variables and lead times, WM-3 attains more accuracy gain for a given level of blurring (or less blurring for a given level of accuracy) than both ENS and AIFS. We present these plots for a wider range of variables and lead times in Appendix A.6.

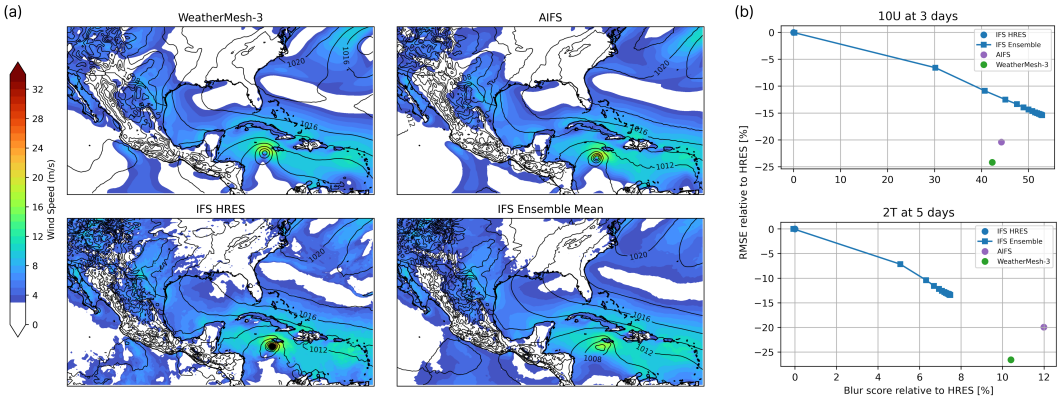

Figure 3: (a) Forecast maps showing 10-meter windspeed and mean sea level pressure at 72 hours lead time from 00z on July, 1, 2024. MLWP models offer notably less fine-grained spatial details, but resolve the intensity of Hurricane Beryl more clearly than the IFS ensemble mean. (b) Blur score vs RMSE for select surface variables. The points on the IFS Ensemble line represent ensemble means for member subsets beginning at 1 member and ending at 51 members.

4 Discussion

4.1 Forecast Accessibility

Compute efficiency The compute efficiency of WM-3 represents a significant step toward democratizing access to high-quality weather forecasts. By reducing hardware requirements by several orders of magnitude compared to traditional NWP systems while maintaining or exceeding their accuracy, WM-3 enables organizations with limited computational resources to run their own customized forecasting systems. A 14-day global forecast can be computed in just 12 seconds on a single RTX 4090, roughly 143,000 times faster (in node seconds per hour lead time) than physics-based numerical weather models (Bodnar et al., 2024). On a single H100 server, hourly global forecasts out to 14 days can be computed in under 10 seconds.

WM-3 runs on hardware as light as a consumer grade laptop with 16GB VRAM and 32GB RAM, which is one third of Aurora's VRAM usage (Bodnar et al., 2024). This accessibility has the potential to benefit entities that previously lacked access to advanced forecasting capabilities due to infrastructure constraints, democratizing access to climate resilience tools.

Modular customization The architecture of WM-3 enables users to ingest additional operational data sources for modular customization. Operational WM-3 currently ingests ECMWF IFS and NOAA GFS analyses via two separate encoders, trained as described in Section 2.2. The architecture is well-suited for users to create modular extensions that ingest additional custom data sources to enhance forecast accuracy. Accessibility to this level of forecast customization will become increasingly pertinent for users in light of rising need for forecasting weather and extreme events.

4.2 FUTURE WORK

Our results suggest several directions for advancing this research. In particular, the computational efficiency of WM-3 makes large ensemble forecasts feasible on modest hardware, opening new possibilities for novel ensemble techniques. These could leverage the model's speed to generate more sophisticated uncertainty quantification and improve forecast reliability. This computational advantage also enables fast integration of additional data sources to improve operational forecasts via a live data assimilation pipeline. This real-time data can be incorporated through additional encoder pathways while maintaining the efficient latent space processing that enables our current performance gains. We are eager to continue future work towards operationalizing fast, accurate, and accessible forecasts.

REFERENCES

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.

Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Anna Vaughan, et al. Aurora: A foundation model of the atmosphere. *arXiv preprint arXiv:2405.13063*, 2024.

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

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

ECMWF. *IFS Documentation CY49R1 - Part V: Ensemble Prediction System*, chapter 5. ECMWF, 11/2024 2024. doi: 10.21957/956d60ad81.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

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

Ali Hassani, Wen-Mei Hwu, and Humphrey Shi. Faster neighborhood attention: Reducing the $o(n^2)$ cost of self attention at the threadblock level. In *Advances in Neural Information Processing Systems*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Proceedings of the platform for advanced scientific computing conference*, pp. 1–11, 2023.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana CA Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, et al. Aifs-ecmwf's data-driven forecasting system. *arXiv preprint arXiv:2406.01465*, 2024a.

Simon Lang, Mihai Alexe, Mariana CA Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D Dueben, Sara Hahner, et al. Aifs-crps: Ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *arXiv preprint arXiv:2412.15832*, 2024b.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*, 2023.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. 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.

Hao-Jun Michael Shi, Tsung-Hsien Lee, Shintaro Iwasaki, Jose Gallego-Posada, Zhijing Li, Kaushik Rangadurai, Dheevatsa Mudigere, and Michael Rabbat. A distributed data-parallel pytorch implementation of the distributed shampoo optimizer for training neural networks at-scale. *arXiv preprint arXiv:2309.06497*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

A Model Details

A.1 Variables

WM-3 forecasts 17 surface variables and 5 atmospheric variables. Each variable is presented in Table 1 below with its corresponding ID from the ECMWF Parameter Database, which contains further information. All atmospheric variables are forecasted at 28 pressure levels: 10, 30, 50, 70, 100, 125, 150, 175, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 875, 900, 925, 950, 975, 1000 hPa.

Variable Name	**ECMWF Parameter ID**
Surface Variables	
10 metre u wind component (10U)	165
10 metre v wind component (10V)	166
2 metre temperature (2T)	167
Mean sea-level pressure (MSL)	151
Total cloud cover	45
2 metre dewpoint temperature	168
100 metre u wind component	246
100 metre v wind component	247
Mean shortwave radiation flux	15
Large-scale precipitation	142
Convective precipitation	143
Maximum 2 metre temperature	201
Minimum 2 metre temperature	202
Large-scale precipitation (6h)	142
Convective precipitation (6h)	143
Maximum 2 metre temperature (6h)	201
Minimum 2 metre temperature (6h)	202
Atmospheric Variables	
Geopotential (Z)	129
Temperature (T)	130
U component of wind (U)	131
V component of wind (V)	132
Specific humidity (Q)	133

Table 1: ECMWF Parameter IDs for WM-3 surface and atmospheric variables.

Most of the surface variables are not available in IFS or GFS analyses, and WM-3 takes as input only the first eight surface variables.

At 0.25 resolution, input and output are arrays of 721×1440. In order to optimize for storage space as well as computational efficiency, we omit the last latitude row (the south pole) and only operate with arrays of 720×1440.

A.2 Model Architecture

The encoder consists of ResNet blocks (He et al., 2016) and convolutional layers. The following constant variables are concatenated with the input: sine and cosine of latitudes and longitutes, the sea-land mask, soil type, topography, and elevation. The encoder includes 2 NATTEN blocks with window size (5, 7, 7) in depth, width, height, corresponding to the vertical height, latitude, and longitude dimensions respectively. The hidden dimension is 1024.

Both processors consist of 10 NATTEN blocks.

The decoder is the reverse of the encoder: it begins with 2 NATTEN blocks, and then a series of ResNet and deconvolutional layers take the output from latent space back to physical space.

To make NATTEN work on a sphere, we implement our own circular padding. At the poles, we use the bump attention behavior from NATTEN. For position encoding of tokens, we use rotary embeddings (Su et al., 2024).

We share our model code in a repository that we open source as part of this submission.

A.3 TRAINING DETAILS

WeatherMesh-3 is pretrained on ERA-5 data from January 23, 1979 to December 28, 2019, as well from February 1, 2021 to December 21, 2023. The year of 2020 is left out of the training set for evaluation as is conventional (Rasp et al., 2024). It is trained for 42,000 steps on a cosine learning rate schedule with a maximum learning rate of 3e-4. The training target is mean squared error (MSE) at target timestep dt, with dt determined by a schedule. The schedule is as follows for the 6-hour processor:

Starting step	**Target dt hours**
0	0, 6, 12
1,000	0, 6, 12, 18, 24
15,000	0, 6, 12, 18, 24, 30
21,000	0, 6, 12, 18, 24, 30, 36
26,000	0, 6, 12, 18, 24, 30, 36, 42
30,000	0, 6, 12, 18, 24, 30, 36, 42, 48

Table 2: Schedule for introducing additional target dts during training.

At each step, a random subset of 5 dts are chosen as training target, while always including the largest dt allowed to make sure the time it takes to step across GPUs is the same.

The model then undergoes another annealing cycle of 16,000 steps training on dts up to 120 hours (5 days). The 1-hour processor, trained with a frozen encoder and decoder, is trained for 25,000 steps on 5 random dts between 0 and 24 at each step.

For operational fine-tuning, we attach two new encoders to take IFS and GFS analyses as input, as shown in Figure 4. We train on IFS and GFS analyses from March 2021 to February 2024.

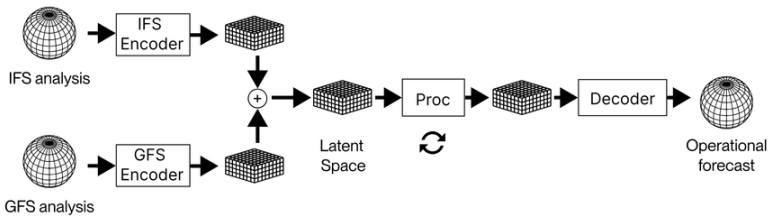

Figure 4: Schematic for operational encoders.

We use the PyTorch implementation of distributed shampoo (Shi et al., 2023; Gupta et al., 2018), a second-order optimizer, and train across 6 RTX 4090 GPUs. We use half precision in training.

A.4 VALIDATION DETAILS

For results in Figure 2, we used a validation period of approximately five months, from March 10, 2024 to July 31, 2024, and evaluated forecasts initialized at 0z. For results in Figure 3, the validation

period is March 21, 2024 to July 31, 2024, again with all forecasts initialized at 0z. The dates are chosen due to the availability of data at the time of writing, not as a result of cherry-picking.

For the accuracy metric, we calculate latitutde-weighted root mean squared error (RMSE)

$$\text{RMSE} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} w(i)(\hat{X}_{i,j}^{t} - X_{i,j}^{t})^2}, \tag{1}$$

where $\hat{X}$ is the predicted value, X is the ground truth value, t ranges over evaluation dates, i and j range over latitude and longitude ($H = 720$ and $W = 1440$ at 0.25 degrees), and $w(i) = \cos\left(\frac{i \cdot \pi}{720}\right)$ is the latitude weighting factor.

For the blur metric, we calculate a blur score

$$\text{Blur score} = \frac{1}{\sqrt{S_{500}}}, \tag{2}$$

where S_{500} is the spectral power at a wavelength of 500 km, as defined in Rasp et al. (2024). The wavelength of 500 km is somewhat arbitrary, but is meant to capture blurring on the scale of e.g. the size of a hurricane without being dominated by smaller artifacts present in a 0.25-degree (25-km) resolution forecast.

A.5 MATEPOINT

A single training sample for weather forecasting requires a lot of computation: one cannot predict weather for only half of the globe at a time, the entire globe must be forecasted at once as it is an interconnected system. Weather over the entire earth is a much larger input than a passage of text in the context window of an LLM or an image. For this reason, we currently train with only a batch size of 1.

It is not possible to train a global weather model without careful considerations of VRAM utilization. Storing the intermediate model activations for a backwards pass for even the shortest forecast step immediately takes hundreds of GiB of VRAM if done naively. As is common practice, we opt to do model checkpoints for the main transformer blocks to dramatically save VRAM at the cost of longer backwards pass compute time. This works, but it breaks down for training longer forecast time horizons.

The challenge with conventional checkpointing is that it requires storing the input to the checkpointed layer. For a transformer, this means storing the latent space for each transformer block. A 6 day forecast requires running over 200 transformer layers, so for a latent space of 200MiB, this quickly takes up 40GiB of VRAM just for these tensors alone. WM-3 was trained entirely on RTX 4090s due to their very attractive cost per compute, and so this would be entirely unworkable.

The solution is an additional step over just checkpointing. Rather than keep the input tensor to a checkpoint on the GPU, we send it back to the CPU to be stored in RAM, a much more abundant resource. Because forecasting longer in time simply means calling more checkpointed transformer layers, this method means that there is zero VRAM cost to longer forecasting during training time, and there is effectively no limit.

We create a forked version of the pytorch checkpoint library named "matepoint" to implement this concept. There are a few computational considerations required in order to do this without slowing down training. First, tensors must be sent back to the GPU on an independent CUDA stream to allow for full parallelization. Second, tensors that are needed during the backwards pass must be pipelined so that they arrive on the GPU before they are needed. If tensors were only moved as needed naively, then this would incur a delay between each transformer layer in the backwards pass while waiting on tensors to arrive. We have also released the library as an open source python package.

A.6 ADDITIONAL RESULTS

We present the RMSE-blur score plots for a wider range of variables in Figures 5–9. The number of IFS ensemble members included to produce the IFS Ensemble line in these RMSE-blur score plots is 1, 2, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, and 51.

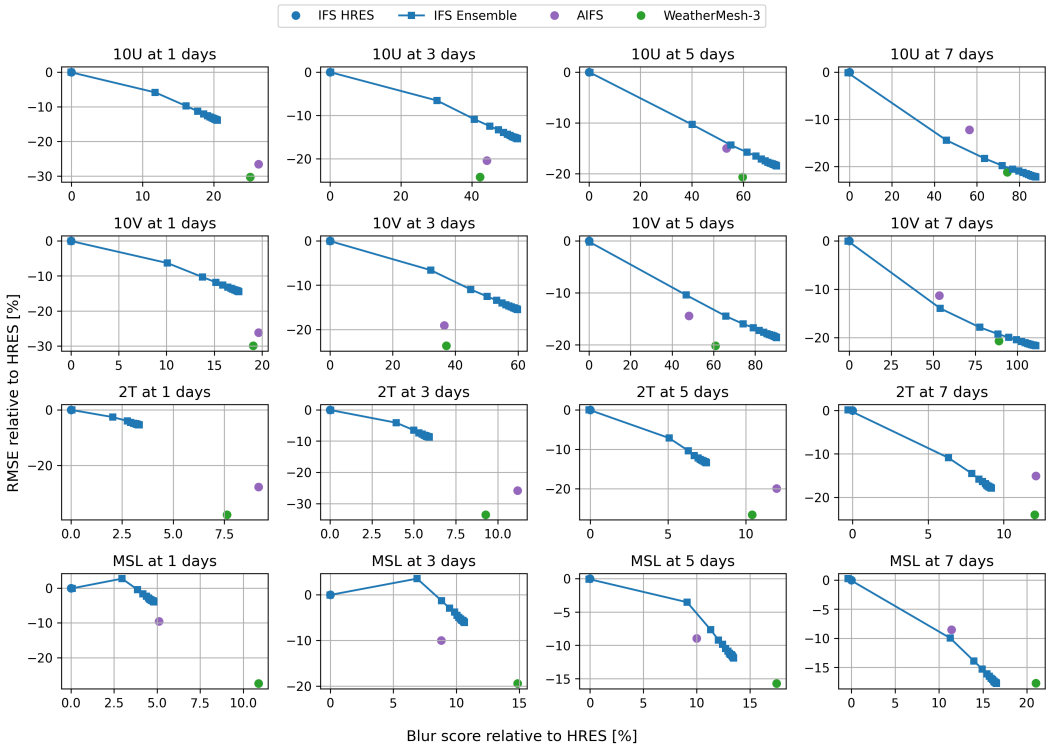

Figure 5: RMSE-blur score for surface variables.

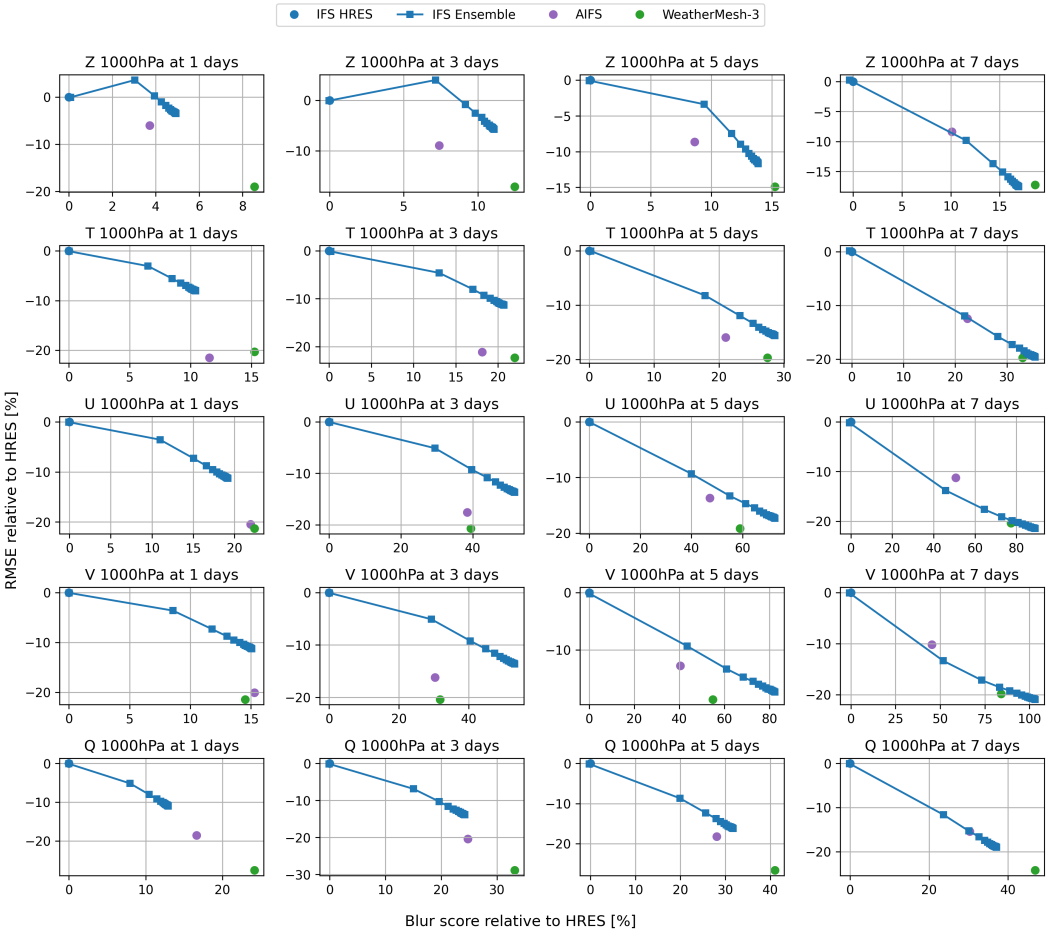

Figure 6: RMSE-blur score for variables at 1000 hPa.

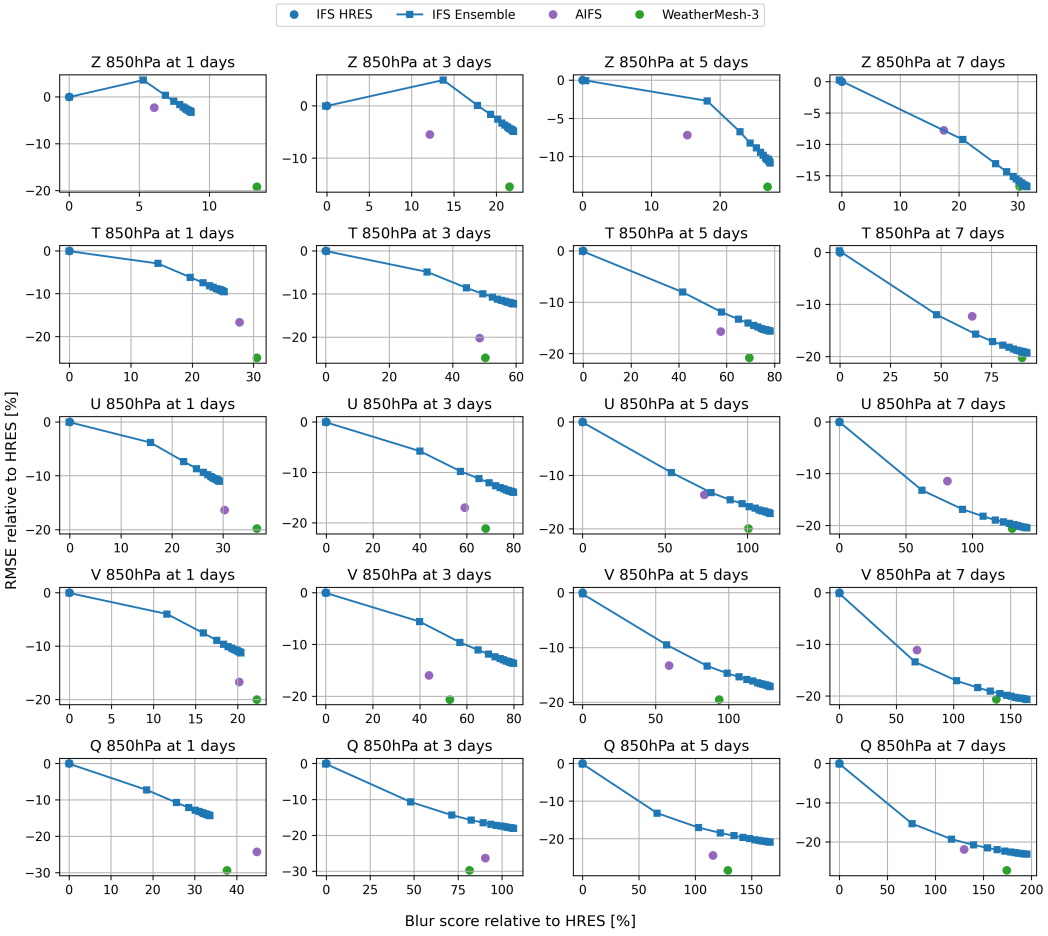

Figure 7: RMSE-blur score for variables at 850 hPa.

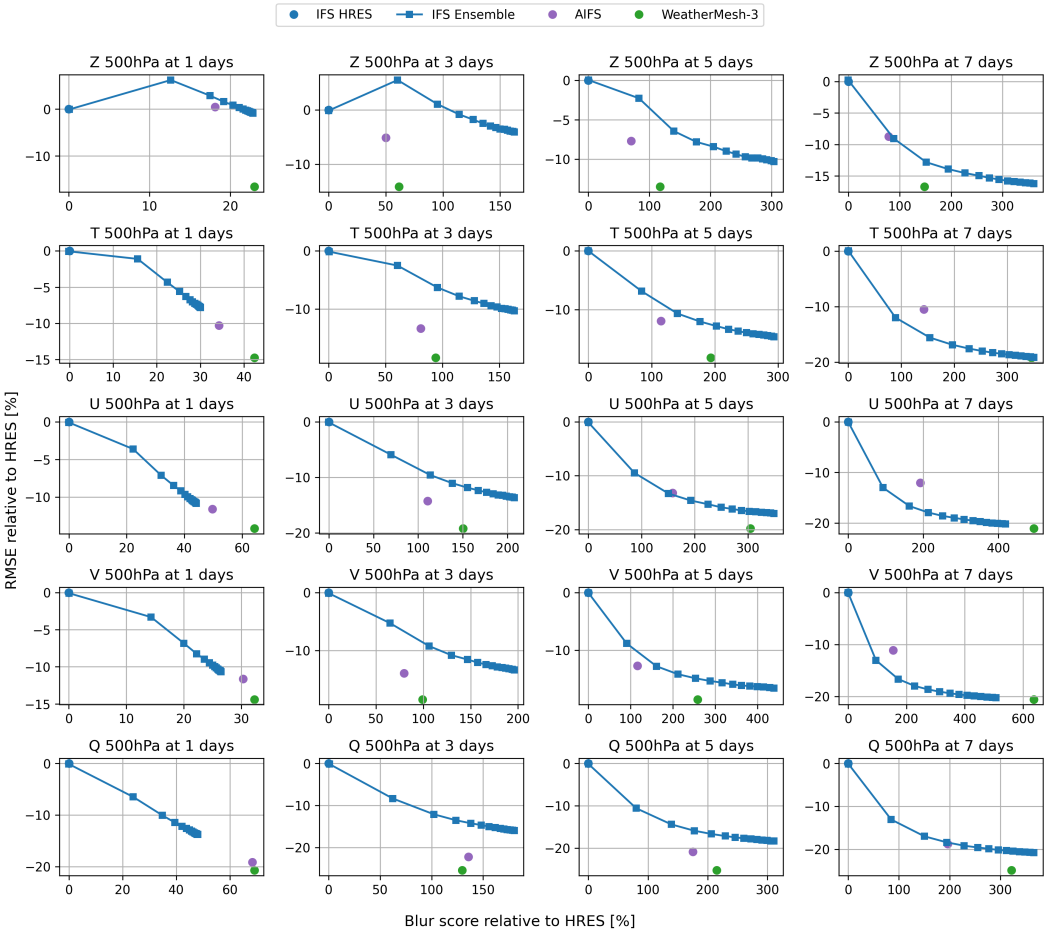

Figure 8: RMSE-blur score for variables at 500 hPa.

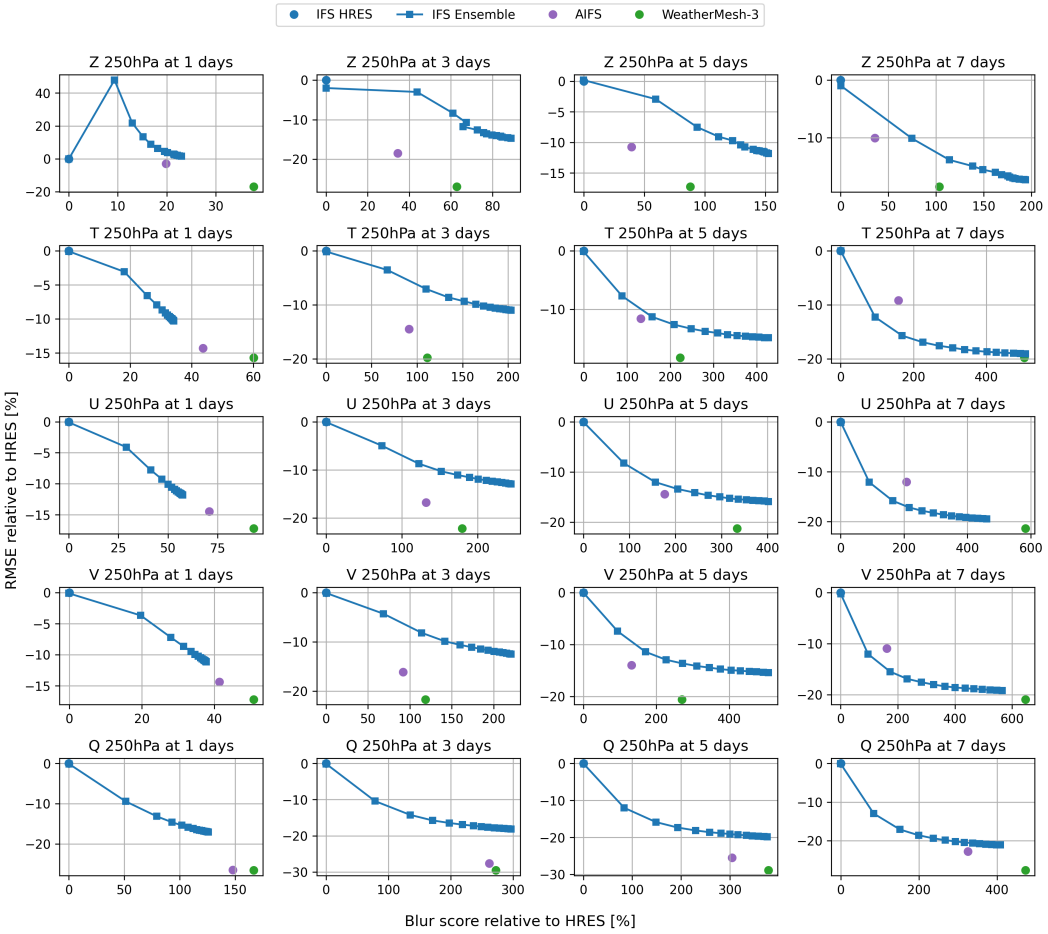

Figure 9: RMSE-blur score for variables at 250 hPa.