

Analyzing and Exploring Training Recipes for Large-Scale Transformer-Based Weather Prediction

JARED D. WILLARD¹,^a PETER HARRINGTON,^a SHASHANK SUBRAMANIAN,^a ANKUR MAHESH,^b
TRAVIS A. O'BRIEN,^{c,d} AND WILLIAM D. COLLINS^{b,d}

^a Lawrence Berkeley National Laboratory, National Energy Research Scientific Computing Center, Berkeley, California

^b Department of Earth and Planetary Science, University of California, Berkeley, Berkeley, California

^c Department of Earth and Atmospheric Sciences, Indiana University Bloomington, Bloomington, Indiana

^d Lawrence Berkeley National Laboratory, Earth & Environmental Sciences Area, Berkeley, California

(Manuscript received 24 July 2024, in final form 25 October 2024, accepted 7 February 2025)

ABSTRACT: The rapid rise of deep learning (DL) in numerical weather prediction (NWP) has led to a proliferation of models which forecast atmospheric variables with comparable or superior skill than traditional physics-based NWP. However, among these leading DL models, there is a wide variance in both the training settings and architecture used. Further, the lack of thorough ablation studies makes it hard to discern which components are most critical to success. In this work, we show that it is possible to attain high forecast skill even with relatively off-the-shelf architectures, simple training procedures, and moderate compute budgets. Specifically, we train a minimally modified Swin Transformer V2 (SwinV2) on ERA5 data and find that it attains superior skill in terms of mean-square errors of deterministic forecasts when compared against the European Centre for Medium-Range Weather Forecasts' Integrated Forecasting System (IFS). Almost all DL-NWP systems share a core set of hyperparameters and design decisions. To aid and expedite future DL-NWP research, we present an in-depth, systematic exploration of different loss functions, model sizes and depths, patch sizes, and multistep training objectives. We also examine the model performance with metrics beyond the typical accuracy (ACC) and RMSE and investigate how the performance scales with model size. Through our open-source code, scoring pipelines, and models, we share our findings on key aspects of the training pipeline. These ablations reduce the necessity for expensive hyperparameter tuning and lower the barrier to entry for future DL-NWP research.

SIGNIFICANCE STATEMENT: This study investigates the potential of using large-scale transformer-based models for weather prediction, showing that it is possible to achieve high forecast accuracy with simpler, off-the-shelf architectures. By training a minimally modified SwinV2 transformer on ERA5 data, we show that the model achieves competitive forecast skill in terms of mean-square error for key variables, outperforming the European Centre for Medium-Range Weather Forecasts' Integrated Forecasting System (IFS) at all lead times. Our findings suggest that effective training strategies, such as multistep fine-tuning and channel-weighted losses, significantly enhance the model's performance. However, we also highlight that these improvements come with trade-offs in other areas, such as ensemble spread and high-frequency spatial detail. This work highlights the promise of deep learning in improving weather forecasts, which could lead to better preparedness and response to weather events, ultimately benefiting society by providing more reliable weather predictions.

KEYWORDS: Forecast verification/skill; Reanalysis data; Software; Deep learning; Machine learning; Neural networks

1. Introduction

As the impacts of climate change continue to grow in severity, it is becoming more important than ever to forecast weather phenomena with high accuracy and fidelity. While operational forecasting has long been performed by numerical weather prediction (NWP) models like the Integrated Forecasting System (IFS), the availability of high-quality reanalysis data (Hersbach et al. 2020) and the onset of advanced deep learning techniques have led to a proliferation of data-driven forecast models in recent years. The power of deep learning in weather prediction has progressed rapidly, producing

models that compete with or outperform leading NWP systems in key forecast metrics (Lam et al. 2023; Bi et al. 2023; Bodnar et al. 2024) only 4 years after the earliest pioneering works (e.g., Dueben and Bauer 2018). This, combined with the fact that deep learning weather models offer unique capabilities that can augment the capabilities of NWP (Ben-Bouallegue et al. 2023), has generated substantial interest in how to design the most effective deep learning approaches for weather prediction.

Recent literature has proposed an abundance of training recipes, network architectures, inference configurations, and compute budgets for the task of global medium-range forecasting. Proposed architectures range from graph neural networks (Keisler 2022; Lam et al. 2023), transformers (Bi et al. 2023; K. Chen et al. 2023; L. Chen et al. 2023; Nguyen et al. 2023b), neural operators (Pathak et al. 2022; Bonev et al. 2023), and convolutional neural networks; the training recipes include different loss functions, normalization methods, time-stepping schemes, and varied sets of atmospheric variables and resolutions/grids.

Willard's current affiliation: National Renewable Energy Laboratory, Golden, Colorado.

Corresponding author: Jared D. Willard, jared.d.willard@gmail.com

For instance, while some works use the standard vision transformer patch embedding (Pathak et al. 2022; Bodnar et al. 2024; Han et al. 2024), others use a space–time cube embedding (Hu et al. 2023; L. Chen et al. 2023), mappings to spherical or icosahedral meshes (Lam et al. 2023; Ramavajjala 2024), or the weather-specific variable tokenization embedding in Nguyen et al. (2023b,a). Loss functions used in these studies include standard and latitude-weighted mean-square error (MSE) (Pathak et al. 2022; Hu et al. 2023; Nguyen et al. 2023a; Rasp et al. 2020; Weyn et al. 2021), mean absolute error (MAE) (Bodnar et al. 2024), and uncertainty-based negative log-likelihood loss (K. Chen et al. 2023). Expanding beyond just deterministic forecasting and conventional deep learning, other works have also explored diffusion models (Price et al. 2023) and hybrid physics–machine learning (ML) schemes (Kochkov et al. 2023). This progress, while exciting, also presents a challenge for researchers, as the differences between models are multifold and sometimes entangled, so separating their effects is not always possible. This is in part due to the high cost of performing hyperparameter tuning on these, mostly vision transformer-based models, being trained on high-resolution datasets with many relevant hyperparameters (Steiner et al. 2021; Yu et al. 2024). While some works present ablations and analyses on parts of their models, it is challenging to do so comprehensively, and there remains a need for analysis under restricted and controlled settings.

With the advent of open benchmarks providing detailed and informative model evaluation (Rasp et al. 2023; Brenowitz et al. 2024), comparisons between and within models are becoming easier, and the recent Stormer model (Nguyen et al. 2023b) and ArchesWeather model studies (Couairon et al. 2024) exemplify what can be done with extensive ablation studies. These works take the standard vision transformer (ViT) architecture and explore the impact of key components of the training pipeline and model architecture on the downstream performance of the model, assessed by the root-mean-square error (RMSE) of deterministic forecasts for key weather variables. Their improvements achieve highly competitive accuracy when compared against other leading models at $\sim 1.4^\circ$ resolution.

In this work, we take a similar approach and aim to assess the effect of different training settings, and the interplay between them, but at full ERA5 resolution (0.25°). The decision to work at full resolution is motivated by the fact that high-resolution forecasts are simply more useful and that excessive blurring and lack of fine-scale detail are current shortcomings of current deep learning–based models (Ben-Bouallegue et al. 2023; Price et al. 2023; Brenowitz et al. 2024). We chose the Swin Transformer V2 (SwinV2) (Liu et al. 2022) architecture as a representative transformer-based architecture that works well at high resolutions. Though other studies have used the SwinV2 backbone within a more complex architecture (L. Chen et al. 2023), here we focus on the relatively “off-the-shelf” version of SwinV2 widely available in deep learning libraries. With SwinV2, we explore several different aspects of the training pipeline. We do not aim to provide a comprehensive analysis of all proposed training strategies but focus on what can achieve good deterministic skill at full ERA5 resolution on 1–5-day

lead times. This focus is partly motivated by observations that multistep fine-tuning and other methods that most improve skill at long lead times exacerbate the issue of blurring, as model predictions tend to resemble more of an ensemble mean rather than a deterministic forecast (Price et al. 2023; Brenowitz et al. 2024). Our contributions are as follows:

- We present a minimally modified SwinV2 model trained on ERA5 at full 0.25° resolution that outperforms IFS in deterministic RMSE skill.
- We present detailed ablations over key training and model settings, finding the effects of latitude weighting, channel weighting, and multistep fine-tuning to be somewhat entangled; they are generally positive but not always constructive or additive with each other.
- We confirm that multistep fine-tuning can improve RMSE but affect sharpness and ensemble spread in transformer architectures as well.

2. Dataset and model details

a. Data

We use the ERA5 (Hersbach et al. 2020) reanalysis dataset, provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) and accessed using the National Center for Atmospheric Research (NCAR) Research Data Archive (RDA) (ECMRWF 2019). The dataset is generated by assimilating a large amount of global observational data using ECMWF’s IFS model. Given its coverage and accuracy, ERA5 data are widely regarded as the most comprehensive and accurate reanalysis archive. ERA5 contains hourly reanalysis data at a spatial resolution of 0.25° (~ 25 km) from years 1979 to the present day. For this study, we subsample ERA5 on $\Delta t = 6$ -h time intervals and select 73 variables from the full dataset to include in the model (this closely follows previous work, e.g., Nguyen et al. 2023b; Bi et al. 2022). These are geopotential height z , winds (u , v), temperature t , and specific humidity q at 13 vertical pressure levels (50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, and 1000 hPa), along with eight single-level/surface variables: surface winds at 10 and 100 m (u_{10} , v_{10} , u_{100} , v_{100}), 2-m temperature (t_{2m}), surface pressure (SP), mean sea level pressure (MSL), and total column water vapor (TCWV). We also include as static additional inputs the land–sea mask, orography, and cosine of zenith angle (indicating time of day yr^{-1}). As mentioned in the previous section, we focus on results at full ERA5 resolution and thus do not downsample the data as in Nguyen et al. (2023b). All data are normalized by the global mean and standard deviation per variable before training. We use years 1979–2015 as well as 2019 for training data and 2016–17 for validation and then evaluate on 2018 in line with other recent deep learning (DL)-based forecast models (Lam et al. 2023; Bi et al. 2022; L. Chen et al. 2023). The 38 years of training data at 6-h time steps yield 55 480 training samples.

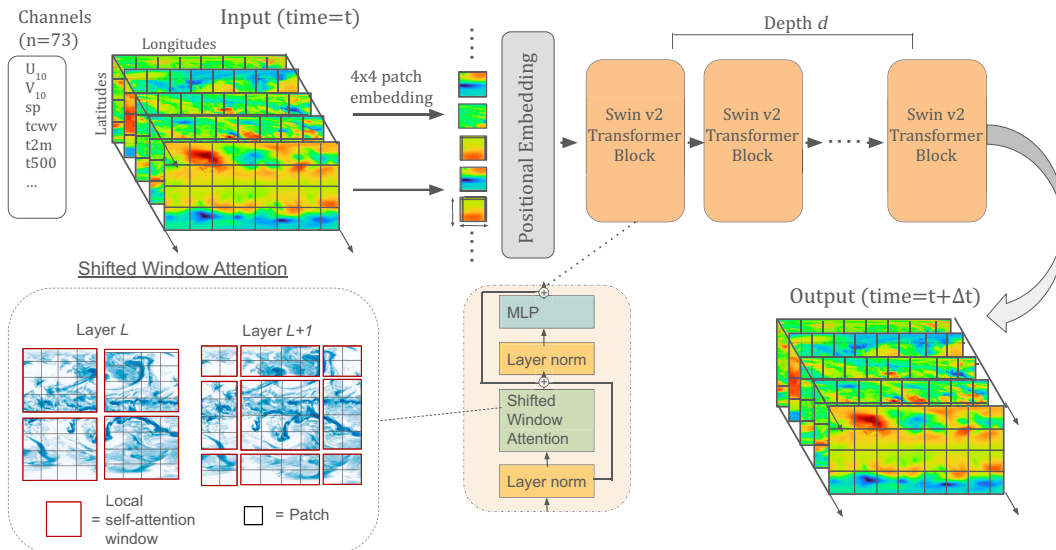


FIG. 1. Schematic of the modified SwinV2 architecture. After normalization, the 73 ERA5 channels first embedded into patches when fed into a positional embedding and then finally through the SwinV2 transformer blocks. Latitude and longitude dimensions are realistically 720×1440 and the attention window size is 9×18 , but these are simplified here for illustrative purposes.

b. Model architecture

We base our model on the SwinV2 (Liu et al. 2022) implementation available in v0.9.2 of the timm library.¹ A model schematic is shown in Fig. 1. While other DL-based forecast models have made extensive modifications to Swin backbones (Bi et al. 2023; L. Chen et al. 2023), we find it sufficient to just minimally modify three aspects of the SwinV2 architecture:

- **Window shifting:** The SwinV2 architecture uses a technique called window attention, where the input data are divided into small, nonoverlapping windows, and the model computes attention within each window. By slightly shifting the windows between different layers of the network, shown as “shifted window attention” in Fig. 1, the model gradually expands its attention to include information from neighboring windows to cover a wider area over multiple layers. The shifting of attention windows is implemented as a torch.roll operation, followed by masking to ensure attention within shifted windows does not cross image boundaries (since 2D images are nonperiodic generally). However, for ERA5 data, the horizontal dimension (longitude) is inherently periodic because it wraps around the globe so we apply masking only along the vertical axis (latitude) to ensure that attention does not cross the boundaries at the poles.
- **Position embedding:** In SwinV2, the position biases within attention windows are carefully defined in a relative coordinate system, to better allow transferring across multiple resolutions/window sizes. In our case, we are restricting ourselves to data at fixed resolution and thus find it sufficient to drop the relative position embeddings in favor of a

standard “absolute” position embedding as in a standard ViT, which is added to the latent space immediately after patch embedding.

- **Nonhierarchical structure:** SwinV2 uses a common vision transformer technique to incorporate a hierarchy across the network layers that sequentially merges patches and decreases resolution to both reduce computational cost and model various spatial scales. We alter the model to be non-hierarchical and maintain the same feature resolution in all layers, which has been shown to be effective for spatiotemporal forecasting in Earth science (Gao et al. 2022).

With these modifications, we train our forecast models in the standard setting, giving the state $\mathbf{X}_t$ as input and predicting the next state $\mathbf{X}_{t+\Delta t}$. In inference, the model is rolled out autoregressively to produce forecasts for lead times larger than Δt . For our baseline SwinV2 model, we use an embedding dimension of 768, a depth of 12 layers, a patch size of 4, 8 attention heads, and a local attention window size of 9×18 . The model is trained with latitude-weighted (described in the following section) $\mathcal{L}_2$ loss using the Adam optimizer with a DropPath rate of 0.1, initial learning rate of 0.001 with a cosine annealing scheduler (Loshchilov and Hutter 2016) (following previous work Pathak et al. 2022), batch size of 64, and trains for 70 epochs on 64 A100 graphics processing units (GPUs) within the Perlmutter high-performance computing system at the National Energy Research Scientific Computing Center. We leverage PyTorch’s Distributed Data Parallel (DDP) (Li et al. 2020) to distribute the training across multiple GPUs efficiently. For the baseline SwinV2 model, the average time for training one epoch was 8 min 40 s and the average time for validating one epoch was 37 s. Total wall clock time is approximately 11 h for the baseline model, which

¹ <https://github.com/huggingface/pytorch-image-models>.

TABLE 1. List of models used to assess model complexity and patch size, along with their corresponding number of parameters. Note that attention windows were also adapted with patch size, leading to a difference in number of parameters between models with the same embedding dimension and depth. We note that the base model number of parameters is maintained in the following experiments adding channel weighting and multistep fine-tuning.

Model name	No. of parameters
swin_depth12_embed768_patch4 (base)	136 666 464
swin_depth12_embed1536_patch4	443 202 144
swin_depth24_embed768_patch4	221 721 024
swin_depth12_embed768_patch8	104 674 656
swin_depth12_embed768_patch16	116 872 032

extends up to 48 h total in the case of the 8-step maximum fine-tuning described in the following section. Seventy epochs were decided based on visual inspection of the low plateau of the validation loss. Activation checkpointing is used as necessary to fit fine-tuning configurations into GPU memory. Modifications and ablations with respect to this baseline model are described in subsequent sections.

c. Ablations and experiments

Model complexity and patch size: In preliminary experimentation, we found that, for a given embedding dimension and depth, the largest possible attention window size performed best, which was an attention window dimension of 9×18 patches. We then probe the effect of larger model sizes in the form of the depth and width dimensions, by building a model with twice as many layers (depth = 24), as well as a model with double the embedding dimension (embed_dim = 1536). We also investigate the three different models with patch sizes of 4, 8, and 16 while keeping the embedding dimension at size 768. For patch sizes of 8 and 16, we adopt the attention window dimensions of 5×10 and 3×6 patches, respectively. This is to account for the increased spatial coverage of each patch affecting the total attention window size. All five of these models and the corresponding number of parameters are listed in Table 1. We note that the rest of the models described in the following experiment will have the same number of parameters as the model with *depth* = 12, *embed_dim* = 768, and *patch_size* = 4.

Channel weighting: Recent work has found it beneficial to carefully weight channels (different weather variables at different vertical levels) in the loss function during training (Nguyen et al. 2023b). In particular, the method first pioneered by Lam et al. (2023) of downweighting as pressure level decreases (higher vertical levels are weighted less) and downweighting according to the standard deviation of temporal differences $\sigma_{\Delta \mathbf{x}}$ has been found to work well. Beyond these physically motivated weights, Lam et al. (2023) also manually impose weights that preferentially emphasize certain surface variables, like t2m, which we adopt as well for consistency. Similar to Nguyen et al. (2023b), we evaluate the effect of this configuration compared against the standard loss where all channels are given equal weight.

We note this experiment is also entangled with “direct” versus “residual” prediction—the channel-weighting method, which

partially weights according to the temporal differences $\sigma_{\Delta \mathbf{x}}$, predicts the difference between the input and target time steps, whereas the standard $\mathcal{L}_2$ loss used in the base models directly predicts the target time step. In this work, we do not explore disentangling these two configurations, but in principle, one could separately apply pressure level/ $\sigma_{\Delta \mathbf{x}}$ and direct/residual prediction.

Multistep fine-tuning: This technique, commonly used in deep learning-based weather prediction models (Pathak et al. 2022; Lam et al. 2023; Bonev et al. 2023; Nguyen et al. 2023b; L. Chen et al. 2023), is designed to enhance long-term forecast accuracy by training the model to make predictions over multiple sequential time steps, rather than focusing solely on the immediate next one. After the initial training phase, where the model learns to predict the immediate next state (e.g., $\mathbf{X}_{t+1}$ from $\mathbf{X}_t$), a fine-tuning phase is introduced. There, the model rolls out its predictions recursively: It uses its own output, such as $\mathbf{X}_{t+1}$, as input to predict further into the future, like $\mathbf{X}_{t+2}$, and continues this process up to the specified forecast horizon. The output at the final time step is then compared against ground truth.

While multistep fine-tuning improves deterministic RMSE, especially at longer lead times, it can also lead to some drawbacks. Predictions can become overly smooth and resemble ensemble means, reducing the forecast’s sharpness and variability (Price et al. 2023; Brenowitz et al. 2024). We evaluate the effects of fine-tuning models with four and eight time step rollouts to build upon the observations found in previous works. This fine-tuning was conducted after the initial 70 training epochs, using a reduced learning rate of 1×10^{-4} for an additional 15 epochs, in line with previous works (Pathak et al. 2022; Guibas et al. 2021). The choice of 15 epochs was based on visual inspection of the stabilization of the validation loss. This approach allows us to explore the balance between improved RMSE and the potential compromise in forecast sharpness and ensemble spread.

Latitude weighting: A large number of works, dating back to the original WeatherBench baseline (Rasp et al. 2020), have used a loss function that is weighted by the cosine of the latitude of each grid cell. This approach compensates for the varying sizes of grid cells on a longitude–latitude grid, where cells near the poles are much smaller than those near the equator. We evaluate the model performance with and without this latitude weighting, although it is applied by default unless otherwise specified.

d. Evaluation

We use the recent open-source Earth-2 Model Intercomparison Project (Earth2Mip) package² (NVIDIA 2023) to score and evaluate models. Scores are averaged over 11 initial conditions evenly spaced throughout 2018, and forecasts are rolled out to 7 days at 6-h intervals. We primarily focus on latitude-weighted deterministic RMSE to compare between different models, but doing so imposes trade-offs with

² <https://github.com/NVIDIA/earth2mip>.

other metrics and desirable aspects of forecast quality (e.g., sharpness/blurriness).

We additionally measure energy spectra and the ensemble spread/skill in a lagged ensemble forecast for some of our models, to further illuminate these trade-offs. Lagged ensembles (Tracton and Kalnay 1993; Branković et al. 1990) provide a parameter-free ensemble forecasting system that can be conveniently used to assess probabilistic performance from deterministic forecasts. We generate the ensemble by treating all forecasts initialized within a specific time window at different initial times as separate ensemble members. We follow the procedure used in Brenowitz et al. (2024), using nine ensemble members initialized 6 h apart which are spread out over a 48-h period and centered on the initialization time for a forecast at a given lead time. Generally, we can expect that models which are more intrinsically more dispersive (i.e., create ensembles with larger spread) to have a larger spread–skill ratio, which should ideally be 1 for an optimal real-world ensemble (Whitaker and Louge 1998).

We also evaluate one of our best performing SwinV2 models on WeatherBench 2 (Rasp et al. 2023), a widely used benchmark for data-driven weather forecasting. Using the deterministic evaluation option, we compare our SwinV2 model using channel weighting, 8-step fine-tuning, and latitude-weighted loss against the IFS_HRES (ECMWF 2023), Pangu-Weather (Bi et al. 2022), and GraphCast (Lam et al. 2023) for forecasts up to 10 days out at 6-h intervals. To match existing WeatherBench 2 scores, we expanded our number of initial times to every 12-h interval in the 2018 evaluation year (731 total initial times).

3. Results

a. Model size, patch size, channel weighting, and multistep fine-tuning

We examine variations of model size and the effect on forecast skill in Fig. 2. While doubling the embedding dimension or depth improves performance generally, the model variant with increased embedding dimension consistently outperforms the deeper model at all lead times. The deeper variant (*swin_depth24_embed768*) struggles to beat the baseline in t2m and u10, and the gap between it and the model with larger embedding dimension becomes as large as 10%–15% in RMSE at 7-day lead times.

Figure 2e shows that a patch size of 4 consistently outperforms for t2m, though it slightly lags behind models with patch sizes of 8 and 16 in later lead times for z500 and u10m.

In Fig. 3, we compare the baseline configuration against a model trained with channel-weighted loss and confirm that the channel weighting seems to improve the model's forecasting accuracy across the majority of lead times. For some variables (e.g., u10m in Fig. 3c), the improvement is more apparent at later lead times. We note the improvement is present even in variables which are actually downweighted by the loss (e.g., z500 in Fig. 3a, which is weighted less since it is further from the surface).

In Fig. 4, we show the effect of multistep fine-tuning and channel weighting simultaneously applied, finding it effective in improving RMSE as previous works have shown. The overall effect seems to be that multistep fine-tuning improves performance most at lead times much larger than the fine-tuning window. For example, up to lead times of ~ 2 –3 days, there is no benefit from fine-tuning up to eight steps (48 h) versus just four steps (24 h)—both improve over the baseline by the same amount. However, at lead times of 5 days and beyond, the gap between the two fine-tuning configurations has increased substantially and the eight-step model performs much better.

b. Downstream effects of multistep fine-tuning

Given the stark improvement in RMSE offered by multistep training, it is worth reiterating that these improvements come at the cost of other qualities desirable in weather models: forecast sharpness and ensemble spread. As a quick demonstration, we show in Fig. 5 the power spectra of model predictions compared against the ground truth ERA5 (averaged over all lead times and initial conditions). The multistep fine-tuned models are clearly deficient in wavenumbers greater than 10 for u10m (Figs. 5a and 5d), indicating blurring. This effect is not present in all variables, as seen in the spectra, e.g., t2m (Figs. 5c and 5f), whose fine-scale power might be more dominated by (static) orographic features like mountain ranges and coastlines rather than dynamics. The spikes and pileup near the Nyquist frequency (Shannon 1949) across the panels in Fig. 5 are caused by the patch embedding in SwinV2. The division into patches introduces discontinuities at the regular interval of the patch size, creating a signal at that frequency in the latent space of the model. Since this signal is above the Nyquist frequency of the latent space, aliasing occurs, causing the observed pileup. Given that the original resolution before patching has a higher Nyquist frequency (half the latitude field dimension or 360), this effect remains apparent in the plots.

The performance of an ensemble constructed from lagged forecasts is shown in Fig. 6. We observe that indeed the spread–skill ratio (Figs. 6d–f) is diminished for both multistep fine-tuned models, confirming the issues presented in Brenowitz et al. (2024). This decrease in spread–skill ratio is not catastrophic, as there still appear to be ensemble skill gains from fine-tuning as shown in the ensemble mean RMSE (Figs. 6a–c) and continuous ranked probability score (CRPS) (Figs. 6g–i). In particular, the 8-step fine-tuned model has better skill at all lead times than the baseline for all three variables in both metrics.

c. Effects of latitude-weighted loss

Though the previous results have all incorporated the latitude-weighted loss, in Table 2, we examine this choice more closely and compare across several model configurations with and without applying latitude weighting. The results are mixed, highlighting the need for caution when drawing conclusions from just one or two experiments alone. For the 8-step fine-tuned model, the latitude-weighted loss is generally helpful

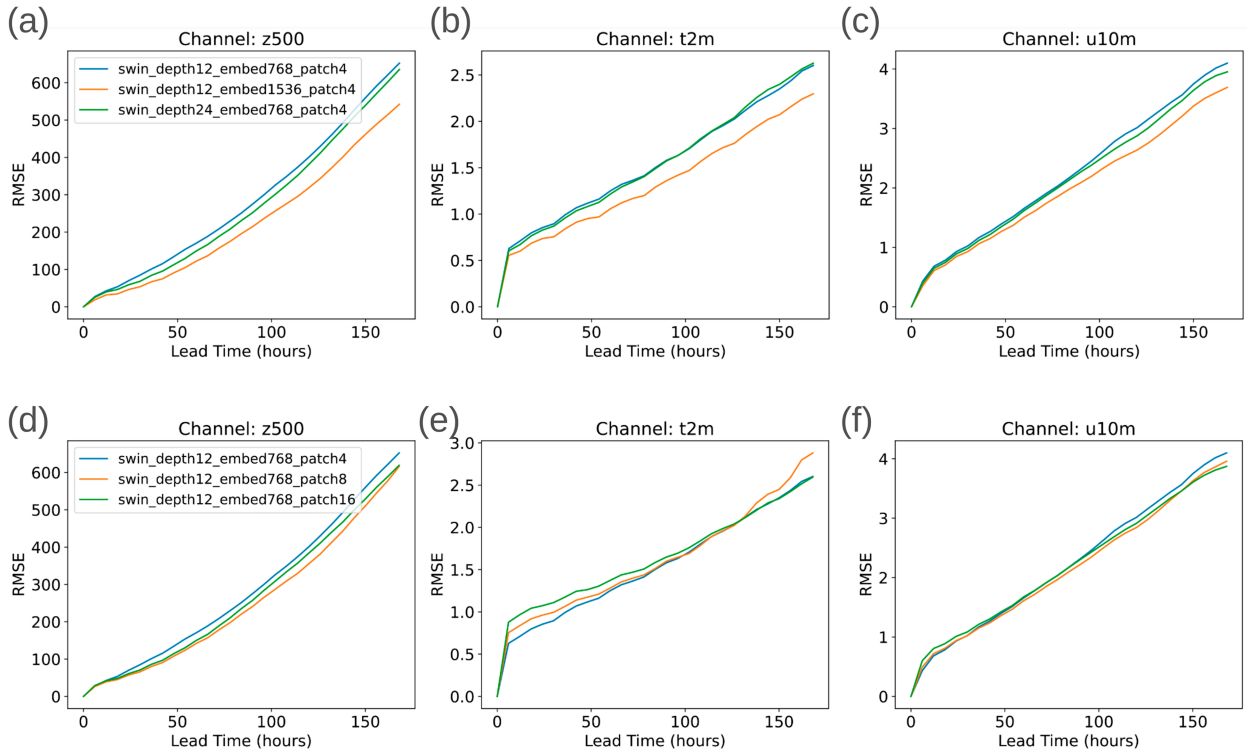


FIG. 2. RMSE comparison of forecasts at lead times up to 7 days comparing model size and patch size for (a),(d) z500, (b),(e) t2m, and (c),(f) u10m. (a)–(c) Comparison of the different model depths and embedding dimensions for the SwinV2 model; (d)–(f) comparison of patch sizes of 4, 8, and 16.

and achieves the lowest RMSE at lead times of 2 and 4 days (at 7 days, the RMSE is roughly the same with and without latitude weighting), with the best overall configuration using both channel weighting and latitude weighting in the loss. Surprisingly, the conclusion is reversed for a model that is only trained over single-step predictions—the latitude-weighted loss does significantly poorer at all lead times, and the best configuration uses

neither channel weighting nor latitude weighting. Thus, the effects of channel weighting, latitude weighting, and multi-step training are entangled; since it is common practice to do more hyperparameter tuning using less expensive configurations first (e.g., tune 1-step training first and then apply multistep training), this can cause problems during the model development process, mainly because tuning hyperparameters

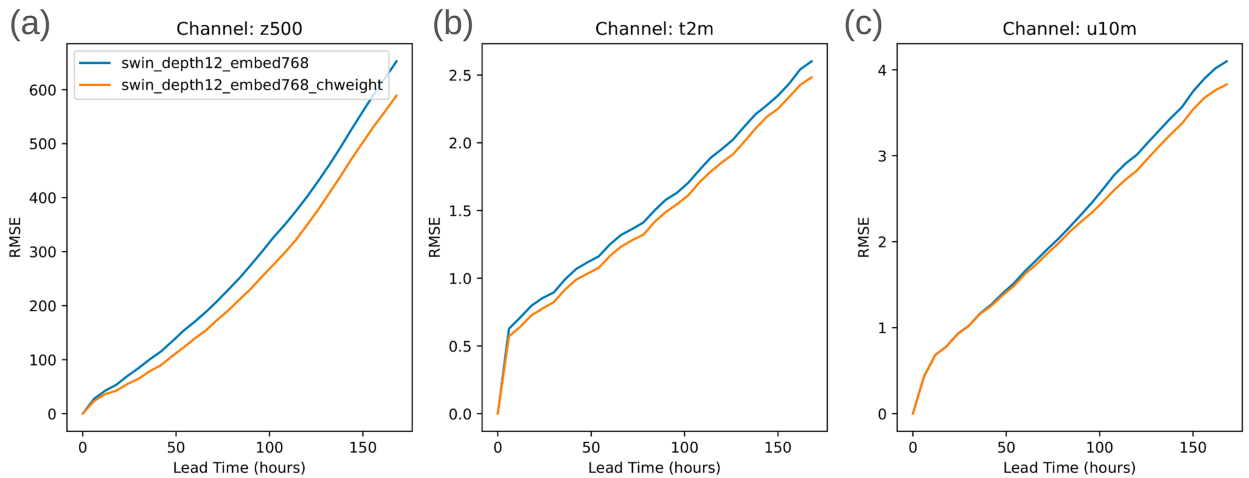


FIG. 3. RMSE comparison of forecasts at lead times up to 7 days for the baseline depth of 12 and embedding dimension of 768 SwinV2 model with and without custom channel weighting for (a) z500, (b) t2m, and (c) u10m.

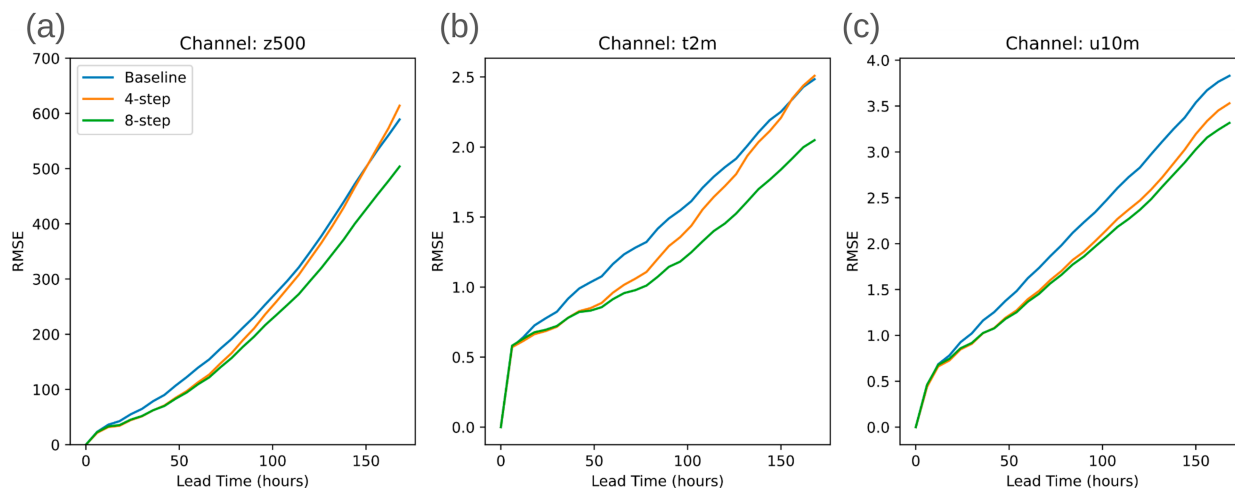


FIG. 4. RMSE comparison of forecasts at lead times up to 7 days for the model trained with custom channel weighting, depth of 12, and embedding dimension of 768 alongside its variants with 4- and 8-step fine-tuning for (a) z500, (b) t2m, and (c) u10m.

for a multistage modeling process quickly becomes very computationally intensive.

d. Additional experiments

Beyond this main set of experiments, we also partially explore other methods proposed in the literature but do not run complete ablation suites due to the initially poor performance we observe. In particular, we attempt to use the variable tokenization embedding (Nguyen et al. 2023b), which uses a cross-attention operation to fuse information between variables after patch embedding. This had previously only been demonstrated on much coarser resolution, and we find that applying it to full 0.25° resolution data poses significant computational challenges due to the memory cost. With activation checkpointing and breaking the operation to run in chunks sequentially, we are able to get it to fit on 80GB A100s, but we observe a slight degradation in the training and validation loss, contrary to the observations of Nguyen et al. (2023b). Since the checkpointing and chunked computation significantly increase the runtime, we do not pursue this further.

We also explore the alternate negative log-likelihood (NLL) loss from K. Chen et al. (2023), which downweights the loss according to a network-predicted uncertainty. Initial experiments under this configuration showed the 1-step RMSE on the validation set lagging behind the baseline configuration, and the predictions were noticeably more blurry. Thus, we did not find this configuration worth including in more extensive ablations.

e. WeatherBench 2 evaluation

The results of the WeatherBench 2 evaluation are shown in Fig. 7. We see that for z500 (Fig. 7a), the SwinV2 model generally outperforms IFS and is on par with Pangu-Weather but slightly worse than GraphCast for lead times less than 7 days, but it surpasses both models beyond 7 days. For t2m (Fig. 7b), the SwinV2 model generally outperformed Pangu-Weather and IFS_HRES but was slightly worse than GraphCast for all

lead times. Last for u10m (Fig. 7c), the model generally outperformed Pangu-Weather at almost all lead times and was similar to GraphCast but slightly worse until about 5 days lead times, beyond which the SwinV2 was more accurate in terms of RMSE. Overall, our SwinV2 model showed strong performance in terms of mean-square error of deterministic forecasts comparable with existing data-driven models that outperform IFS_HRES, especially for forecasts extending beyond 5–7 days for z500 and u10m.

4. Discussion

In an era of abundant high-complexity ad hoc models that may be inaccessible to many researchers, it is important to understand the power of off-the-shelf models and perform ablation to identify which of the multitude of new techniques is effective. This understanding is vital to the advancement of the nascent and rapidly advancing discipline of deep learning-based weather prediction. This work demonstrates that a relatively off-the-shelf SwinV2 transformer architecture, in terms of mean-square error of deterministic forecasts, can outperform IFS and compete with other leading deep learning models with the proper training procedure. Multiple ablations were tested and analyzed, covering model depth, embedding dimension, patch size, latitude weighting, channel weighting, and multistep fine-tuning. We see increased embedding dimension improved model performance compared to increased depth in a model, and this could be due to the greater increase in number of parameters (Table 1). Similar results were reported for the Aurora model (Bodnar et al. 2024) where increasing model size improved validation performance, though we note these results are based on single-step validation forecast error rather than the longer lead time rollouts of up to 12 days employed in this study. We also see that custom channel weighting improved performance on several key weather variables across the majority of tested lead times, mirroring the findings in Nguyen et al. (2023b) at coarser resolution and

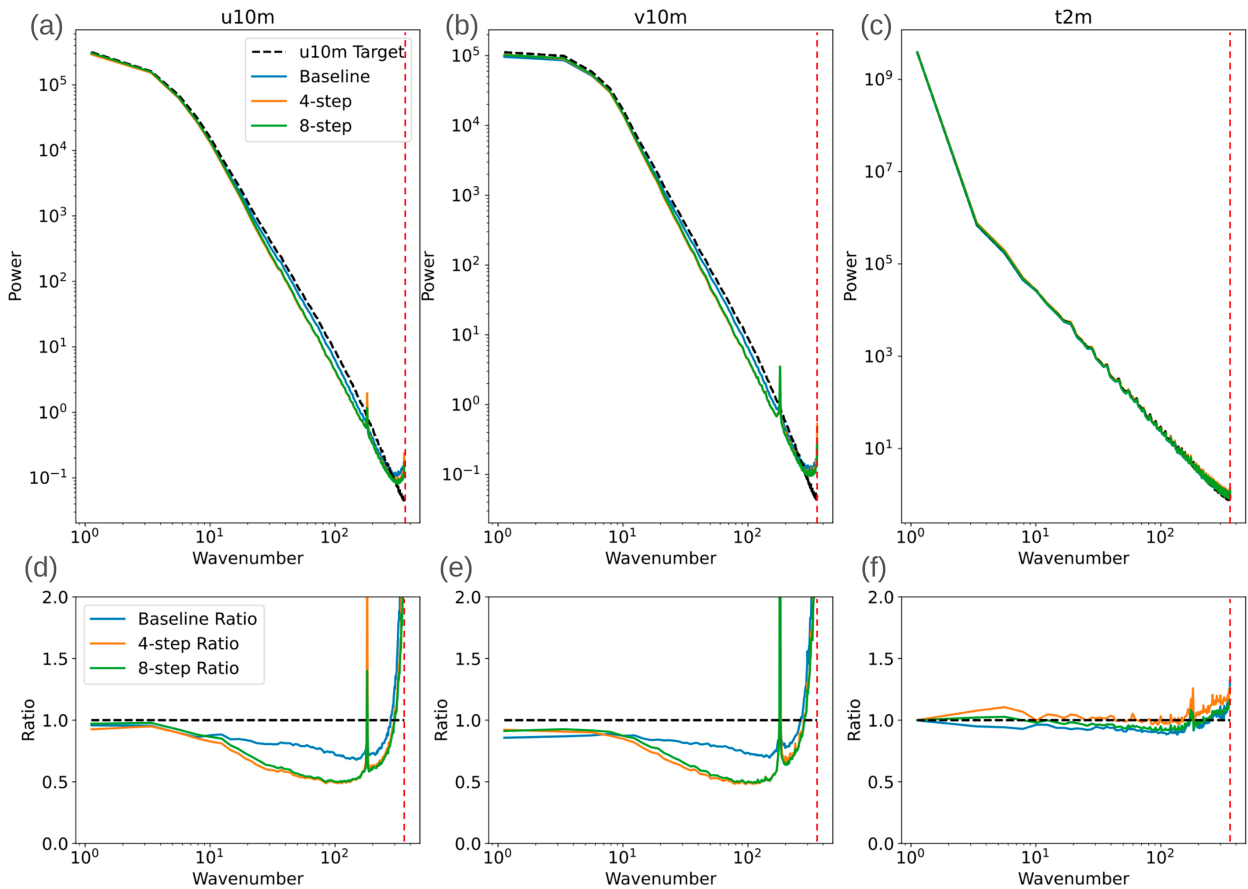


FIG. 5. Spatial frequency representation for the (a),(d) u10m, (b),(e) v10m, and (c),(f) t2m variables across three model configurations: the baseline model using channel weighting and two fine-tuned versions of that model using 4-step and 8-step fine-tuning. For each variable and model, (a)–(c) shows the power spectral density (PSD) of both the target ERA5 (black line) and prediction (red line) on a logarithmic scale. (d)–(f) The ratio of predicted to actual PSD values, with a ratio of 1 (dashed line) indicating perfect alignment. Ratios above or below 1 indicate overestimations or underestimations of power at specific spatial frequencies, respectively. The Nyquist frequency (half the latitude dimension, 360) is shown by the dotted red vertical lines, representing the highest wavenumber that can be resolved by the grid.

indicating pressure-weighted loss and weighting surface variables higher are still beneficial at higher resolutions. We note that this may not be the optimal weighting, but the clear improvement in multiple works indicates that it is worthy of consideration in future model development. Interestingly, the observed improvement in performance for variables like z500, which are downweighted by the channel-weighted loss, might seem counterintuitive. One possible explanation is that downweighting variables with large time step-to-time step changes allows the optimizer to focus on more predictable components of the system. This could indirectly benefit midlevel variables like z500 by reducing the influence of highly variable, less predictable upper-atmosphere data. Additionally, optimizing for more stable features might create a balanced representation across atmospheric layers, indirectly improving variables like z500 that are often indicators of broader atmospheric dynamics. The latitude-weighted loss function has a mixed effect depending on the other parameters and lead times, indicating it may not be necessary for good performance

to emphasize the larger grid cells closer to the equator. However, multistep fine-tuning emerged as a critical strategy, enabling the model to significantly improve performance over longer forecast lead times by considering the model performance over multiple time steps of autoregressive rollout during training. This approach improves the deterministic forecast skill but introduces trade-offs in forecast sharpness and ensemble spread, necessitating additional stabilization methods to maintain overall forecast quality. The variable tokenization layer (Nguyen et al. 2023b) and negative log-likelihood loss function (K. Chen et al. 2023) were found to be largely ineffective for this experimental setup, though it is possible that the variable tokenization layer could be more effective at lower resolutions, as shown in its original study.

Compared with other state-of-the-art deterministic forecasting models like GraphCast and Pangu-Weather, our model demonstrates competitive performance and even outperforms these models in predicting key weather variables such as geopotential height (z500) and 10-m wind

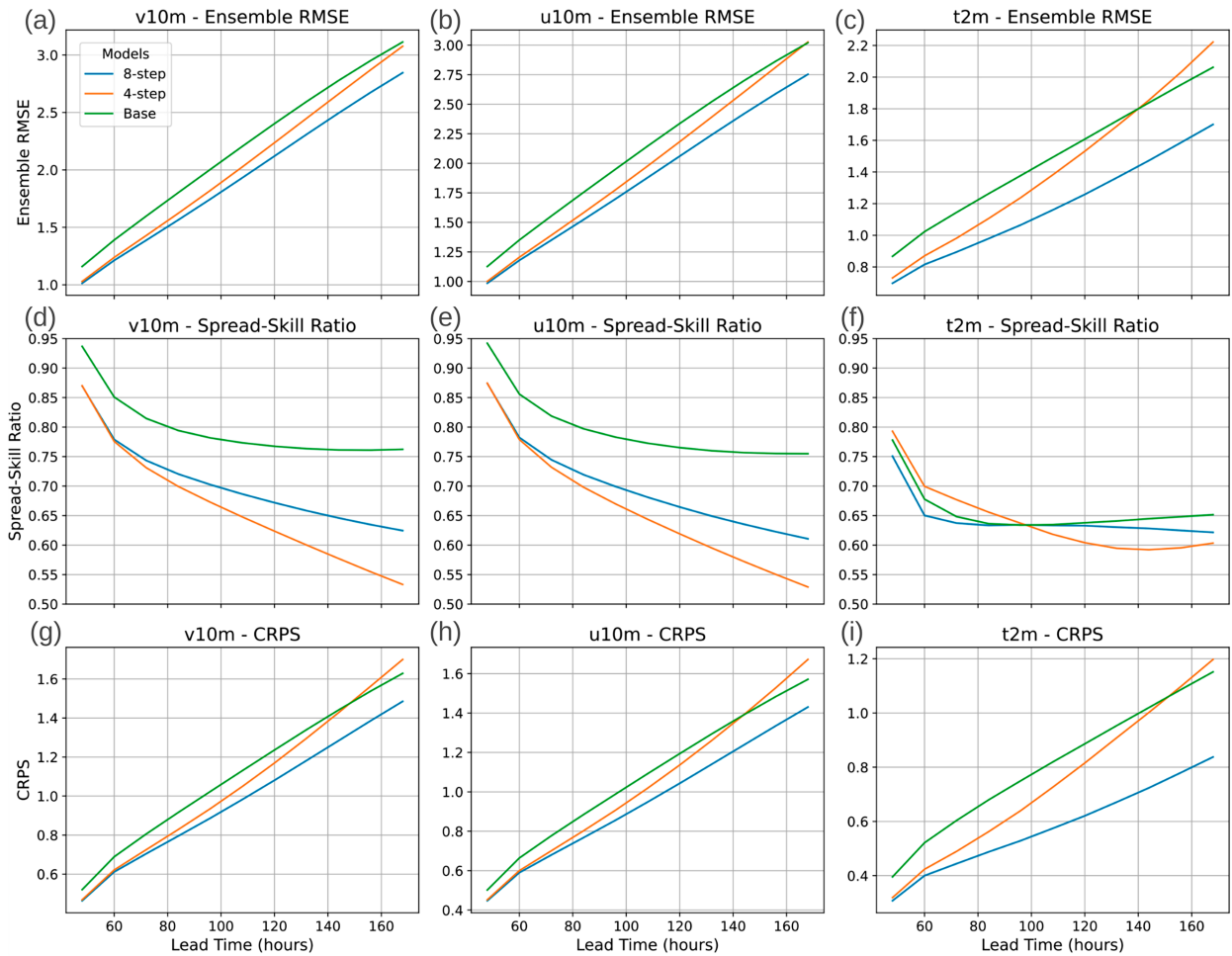


FIG. 6. Lagged ensemble forecast performance for v10, u10, and t2m across three model configurations: the baseline model using channel weighting and two fine-tuned versions of that model using 4-step and 8-step fine-tuning. (a)–(c) The ensemble mean RMSE calculated over the nine-member ensemble, (d)–(f) the spread skill calculated by dividing the standard deviations of predictions across the ensemble by the ensemble RMSE, and (g)–(i) the CRPS calculated across the ensemble.

speed (u10m) at longer lead times. We note that GraphCast does 12-step fine-tuning beyond our just 8 step and believe our model accuracy would likely improve at longer lead times if fine-tuning was further expanded to 12 step. Overall, we believe our approach has significant potential

for refinement and enhancement, especially given its current competitive standing against leading models in the field. Any further improvements to the general SwinV2 model architecture will also likely improve this model as well.

TABLE 2. RMSE for Z500 (m) at 2-, 4-, and 7-day forecast lead times for different model configurations trained with and without latitude-weighted loss. All models have a depth of 12 and an embedding dimension of 768. The best results for 1-step and 8-step training are highlighted in bold.

Channel weighting	No. of step training	Latitude weighted	Z500 (2 days)	Z500 (4 days)	Z500 (7 days)
—	1	—	95.74	237.36	549.18
—	1	✓	134.29	299.96	652.52
✓	1	—	102.31	261.08	572.33
✓	1	✓	106.94	253.72	589.08
—	8	—	85.48	218.65	503.28
—	8	✓	84.01	217.19	507.65
✓	8	—	88.48	227.77	505.18
✓	8	✓	83.20	216.66	503.93

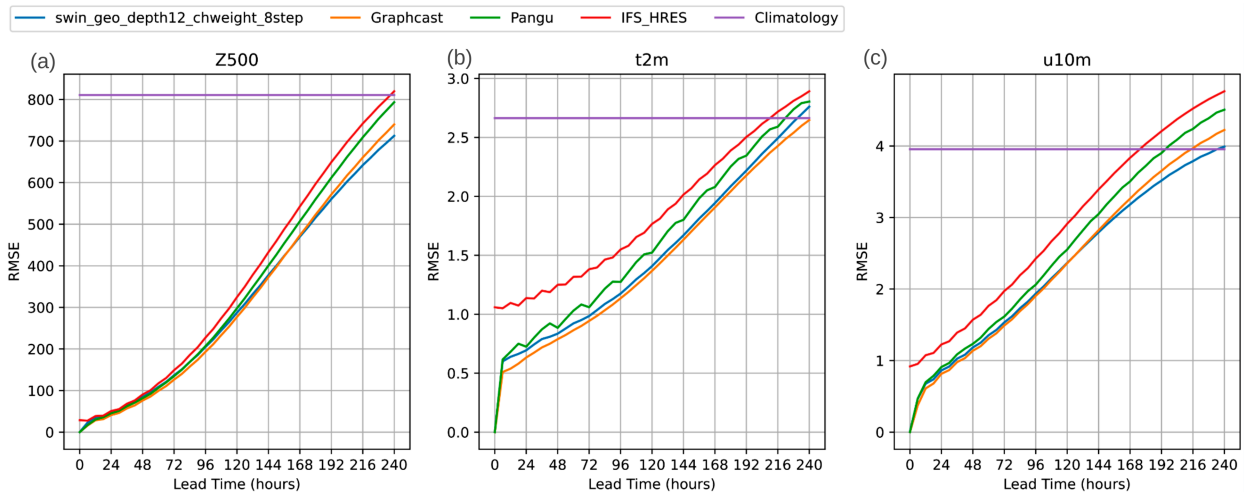


FIG. 7. WeatherBench 2 deterministic RMSE comparison of forecasts of (a) z500, (b) t2m, and (c) u10m at lead times up to 10 days for the SwinV2 model using channel weighting, 8-step fine-tuning, and latitude-weighted loss, IFS_HRES, Pangu-Weather, and GraphCast compared to climatology.

In conclusion, even though we cannot include all ablations on techniques from the multitude of recent works in a single study, our findings highlight that relatively simple modifications to an off-the-shelf model, rather than complex custom architectures, can still lead to competitive performance in terms of RMSE. We also acknowledge that these ablations are performed on a transformer-based architecture and some results may not extend to graph neural-network-based architectures [e.g., GraphCast, Artificial Intelligence Forecasting System (AIFS); Lang et al. 2024], though GraphCast has shown the same channel weighting and a similar multistep fine-tuning. We also note the promising direction could be exploring beyond the rectilinear grid to a spherical or Hierarchical Equal Area Isolatitude Pixelization (HEALPix) (Gorski et al. 2005) mesh to address issues like disproportionate compute being used at the grid convergence near the poles (Ramavajjala 2024; Karlbauer et al. 2024). Overall, we believe our model can serve as a solid foundation for further advancements in deep learning-based weather forecasting, with opportunities to enhance its scalability and reliability through increased embedding dimensions, ensemble-based approaches, or hybrid physics–ML methods. Given that our models are accessible and trained with moderate compute budgets, we believe they are prime candidates for creating both increased parameter models and large model ensembles and hope they will be of great practical use to the community overall.

Acknowledgments. This research was supported by the Director, Office of Science, Office of Biological and Environmental Research of the U.S. Department of Energy under Contract DE-AC02-05CH11231 and the Regional and Global Model Analysis Program area within Earth and Environmental Systems Modeling Program. The research used resources of the National Energy Research Scientific Computing Center (NERSC), also supported by the Office of

Science of the U.S. Department of Energy, under Contract DE-AC02-05CH11231. The computation for this paper was supported in part by the DOE Advanced Scientific Computing Research (ASCR) Leadership Computing Challenge (ALCC) 2023-2024 award “Huge Ensembles of Weather Extremes using the Fourier Forecasting Neural Network” to William Collins (LBNL).

Data availability statement. The primary code used in this study is available at https://github.com/NERSC/swin_v2_weather/. The repository README also has links to ERA5 data, neural network model weights, and precomputed statistics for normalization. The version of Earth2Mip used that contains the SwinV2 transformer implementation can be found at <https://github.com/jdwillard19/earth2mip-swin-fork>.

REFERENCES

- Ben-Bouallegue, Z., and Coauthors, 2023: The rise of data-driven weather forecasting. arXiv, 2307.10128v2, <https://doi.org/10.48550/arXiv.2307.10128>.
- Bi, K., L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, 2022: Pangu-Weather: A 3D high-resolution model for fast and accurate global weather forecast. arXiv, 2211.02556v1, <https://doi.org/10.48550/arXiv.2211.02556>.
- , —, —, —, —, and —, 2023: Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, **619**, 533–538, <https://doi.org/10.1038/s41586-023-06185-3>.
- Bodnar, C., and Coauthors, 2024: A foundation model for the Earth system. arXiv, 2405.13063v3, <https://doi.org/10.48550/arXiv.2405.13063>.
- Bonev, B., T. Kurth, C. Hundt, J. Pathak, M. Baust, K. Kashinath, and A. Anandkumar, 2023: Spherical Fourier neural operators: Learning stable dynamics on the sphere. arXiv, 2306.03838v1, <https://doi.org/10.48550/arXiv.2306.03838>.
- Branković, Č., T. N. Palmer, F. Molteni, S. Tibaldi, and U. Cubasch, 1990: Extended-range predictions with ECMWF

- models: Time-lagged ensemble forecasting. *Quart. J. Roy. Meteor. Soc.*, **116**, 867–912, <https://doi.org/10.1002/qj.49711649405>.
- Brenowitz, N. D., and Coauthors, 2024: A practical probabilistic benchmark for AI weather models. arXiv, 2401.15305v2, <https://doi.org/10.48550/arXiv.2401.15305>.
- Chen, K., and Coauthors, 2023: FengWu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. arXiv, 2304.02948v1, <https://doi.org/10.48550/arXiv.2304.02948>.
- Chen, L., X. Zhong, F. Zhang, Y. Cheng, Y. Xu, Y. Qi, and H. Li, 2023: FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. *npj Climate Atmos. Sci.*, **6**, 190, <https://doi.org/10.1038/s41612-023-00512-1>.
- Couairon, G., C. Lessig, A. Charantonis, and C. Monteleoni, 2024: ArchesWeather: An efficient AI weather forecasting model at 1.5° resolution. arXiv, 2405.14527v2, <https://doi.org/10.48550/arXiv.2405.14527>.
- Dueben, P. D. and P. Bauer, 2018: Challenges and design choices for global weather and climate models based on machine learning. *Geosci. Model Dev.*, **11**, 3999–4009, <https://doi.org/10.5194/gmd-11-3999-2018>.
- ECMRWF, 2019: ERA5 reanalysis (0.25 degree latitude-longitude grid). UCAR/NCAR—Research Data Archive, accessed 22 August 2024, <https://doi.org/10.5065/BH6N-5N20>.
- , 2023: Atmospheric model high resolution 10-day forecast (Set I—HRES). Accessed 5 April 2024, <https://platform.destine.eu/collection/ecmwf-data/ecmwf-real-time-forecasts/ecmwf-real-time-forecasts-open-data/atmospheric-model-high-resolution-10-day-forecast-set-i-hres/>.
- Gao, Z., X. Shi, H. Wang, Y. Zhu, Y. B. Wang, M. Li, and D.-Y. Yeung, 2022: Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, Curran Associates Inc., 25 390–25 403, https://proceedings.neurips.cc/paper_files/paper/2022/hash/a2affd71d15e8fedffe18d0219f4837a-Abstract-Conference.html.
- Gorski, K. M., E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann, 2005: HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *Astrophys. J.*, **622**, 759–771, <https://doi.org/10.1086/427976>.
- Guibas, J., M. Mardani, Z. Li, A. Tao, A. Anandkumar, and B. Catanzaro, 2021: Adaptive Fourier Neural Operators: Efficient token mixers for transformers. arXiv, 2111.13587v2, <https://doi.org/10.48550/arXiv.2111.13587>.
- Han, T., and Coauthors, 2024: FengWu-GHR: Learning the kilometer-scale medium-range global weather forecasting. arXiv, 2402.00059v1, <https://doi.org/10.48550/arXiv.2402.00059>.
- Hersbach, H., and Coauthors, 2020: The ERA5 global reanalysis. *Quart. J. Roy. Meteor. Soc.*, **146**, 1999–2049, <https://doi.org/10.1002/qj.3803>.
- Hu, Y., L. Chen, Z. Wang, and H. Li, 2023: SwinVRNN: A data-driven ensemble forecasting model via learned distribution perturbation. *J. Adv. Model. Earth Syst.*, **15**, e2022MS003211, <https://doi.org/10.1029/2022MS003211>.
- Karlbauer, M., N. Cresswell-Clay, D. R. Durran, R. A. Moreno, T. Kurth, B. Bonev, N. Brenowitz, and M. V. Butz, 2024: Advancing parsimonious deep learning weather prediction using the HEALPix mesh. *J. Adv. Model. Earth Syst.*, **16**, e2023MS004021, <https://doi.org/10.1029/2023MS004021>.
- Keisler, R., 2022: Forecasting global weather with graph neural networks. arXiv:2202.07575v1, <https://doi.org/10.48550/arXiv.2202.07575>.
- Kochkov, D., and Coauthors, 2023: Neural general circulation models for weather and climate. arXiv, 2311.07222v3, <https://doi.org/10.48550/arXiv.2311.07222>.
- Lam, R., and Coauthors, 2023: Learning skillful medium-range global weather forecasting. *Science*, **382**, 1416–1421, <https://doi.org/10.1126/science.adf2336>.
- Lang, S., and Coauthors, 2024: AIFS—ECMWF’s data-driven forecasting system. arXiv, 2406.01465v2, <https://doi.org/10.48550/arXiv.2406.01465>.
- Li, S., and Coauthors, 2020: Pytorch distributed: Experiences on accelerating data parallel training. arXiv, 2006.15704v1, <https://doi.org/10.48550/arXiv.2006.15704>.
- Liu, Z., and Coauthors, 2022: Swin transformer v2: Scaling up capacity and resolution. *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New Orleans, LA, Institute of Electrical and Electronics Engineers, 12 009–12 019, <https://doi.org/10.1109/CVPR52688.2022.01170>.
- Loshchilov, I., and F. Hutter, 2016: SGDR: Stochastic gradient descent with warm restarts. arXiv, 1608.03983v5, <https://doi.org/10.48550/arXiv.1608.03983>.
- Nguyen, T., J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover, 2023a: ClimaX: A foundation model for weather and climate. arXiv, 2301.10343v1, <https://doi.org/10.48550/arXiv.2301.10343>.
- , and Coauthors, 2023b: Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. arXiv, 2312.03876v2, <https://doi.org/10.48550/arXiv.2312.03876>.
- NVIDIA, 2023: Earth-2 Model Intercomparison Project (MIP). GitHub, accessed 22 April 2024, <https://github.com/NVIDIA/earth2mip>.
- Pathak, J., and Coauthors, 2022: Fourcastnet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. arXiv, 2202.11214v1, <https://doi.org/10.48550/arXiv.2202.11214>.
- Price, I., and Coauthors, 2023: Gencast: Diffusion-based ensemble forecasting for medium-range weather. arXiv, 2312.15796v2, <https://doi.org/10.48550/arXiv.2312.15796>.
- Ramavajjala, V., 2024: HEAL-ViT: Vision transformers on a spherical mesh for medium-range weather forecasting. arXiv, 2403.17016v1, <https://doi.org/10.48550/arXiv.2403.17016>.
- Rasp, S., P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, 2020: WeatherBench: A benchmark data set for data-driven weather forecasting. *J. Adv. Model. Earth Syst.*, **12**, e2020MS002203, <https://doi.org/10.1029/2020MS002203>.
- , and Coauthors, 2023: Weatherbench 2: A benchmark for the next generation of data-driven global weather models. arXiv, 2308.15560v2, <https://doi.org/10.48550/arXiv.2308.15560>.
- Shannon, C. E., 1949: Communication in the presence of noise. *Proc. IRE*, **37**, 10–21, <https://doi.org/10.1109/JRPROC.1949.232969>.
- Steiner, A., A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, 2021: How to train your ViT? data, augmentation, and regularization in vision transformers. arXiv, 2106.10270v2, <https://doi.org/10.48550/arXiv.2106.10270>.
- Tracton, M. S. and E. Kalnay, 1993: Operational ensemble prediction at the National Meteorological Center: Practical aspects. *Wea. Forecasting*, **8**, 379–398, [https://doi.org/10.1175/1520-0434\(1993\)008<0379:OEPATN>2.0.CO;2](https://doi.org/10.1175/1520-0434(1993)008<0379:OEPATN>2.0.CO;2).

- Weyn, J. A., D. R. Durran, R. Caruana, and N. Cresswell-Clay, 2021: Sub-seasonal forecasting with a large ensemble of Deep-Learning Weather Prediction models. *J. Adv. Model. Earth Syst.*, **13**, e2021MS002502, <https://doi.org/10.1029/2021MS002502>.
- Whitaker, J. S., and A. F. Loughe, 1998: The relationship between ensemble spread and ensemble mean skill. *Mon. Wea. Rev.*, **126**, 3292–3302, [https://doi.org/10.1175/1520-0493\(1998\)126<3292:TRBESA>2.0.CO;2](https://doi.org/10.1175/1520-0493(1998)126<3292:TRBESA>2.0.CO;2).
- Yu, S., P.-L. Ma, B. Singh, S. Silva, and M. Pritchard, 2024: Two-step hyperparameter optimization method: Accelerating hyperparameter search by using a fraction of a training dataset. *Artif. Intell. Earth Syst.*, **3**, e230013, <https://doi.org/10.1175/AIES-D-23-0013.1>.