Comparative study of Linear Attention Architectures for Geopotential Height Forecasting

Varuni Sastry[1][0009-0007-0366-4186] and Vishwas Rao[1][0000-0002-4395-6075]

Argonne National Laboratory, Lemont, USA {vsastry,vhebbur}@anl.gov

Abstract. Accurate forecasting of geopotential height is crucial for numerical weather prediction and climate modeling. While transformer-based models have shown promising results, their quadratic computational complexity in sequence length limits their scalability to long temporal contexts. In this work, we present a comprehensive comparison of linear attention architectures—including Gated Linear Attention (GLA), DeltaNet, Gated DeltaNet, Kimi Delta Attention (KDA), and Mamba2—for geopotential height forecasting on NCEP reanalysis data. We evaluate these architectures against full-attention transformers in terms of forecasting accuracy, computational efficiency, and memory usage. Our experiments demonstrate that linear attention mechanisms achieve competitive forecasting performance while offering significant computational advantages for processing long temporal sequences.

Keywords: Linear Attention · Weather Forecasting · Geopotential Height · State Space Models · Deep Learning

1 Introduction

Geopotential height, is a fundamental variable in atmospheric science, representing the height of a pressure surface above mean sea level and serving as a critical indicator for weather forecasting [6]. Accurate prediction of geopotential height fields enables identification of synoptic-scale features such as troughs, ridges, and jet streams, that influence surface weather. Traditional numerical weather prediction (NWP) solves complex partial differential equations governing atmospheric dynamics, requires substantial compute and careful parameterization of sub-grid processes [1].

Recent advances in deep learning have demonstrated that data-driven approaches can achieve competitive forecasting skill while significantly reducing computational costs at inference time [15,2,9]. Among these approaches, transformer architectures [20] have emerged as a powerful paradigm for modeling spatio-temporal dependencies in weather data. However, the self-attention mechanism's $\mathcal{O}(n^2)$ complexity with respect to sequence length poses significant challenges for capturing long-range temporal dependencies inherent in atmospheric processes.

Linear attention mechanisms offer a promising solution to this scalability challenge by reducing the complexity to $\mathcal{O}(n)$, enabling efficient processing of

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

longer sequences [8]. Recent innovations in this space include Gated Linear Attention (GLA) [22], which introduces data-dependent decay with gating mechanisms; DeltaNet [18] and its variant Gated-DeltaNet [21], which applies the delta rule from associative memory; Mamba-2 evolves from the original Mamba state space model by simplifying and restructuring the state dynamics computation introducing scalar identities and head dimensions, making it both algorithmically and hardware-efficient [4,3]. Most recently, Moonshot AI introduced Kimi Delta Attention (KDA) [27], which combines channel-wise gating with hardware-aware chunked recurrence for improved efficiency.

In this paper, we present a systematic comparison of the linear attention architectures-namely, GLA, DeltaNet, Gated-DeltaNet, Mamba2, and KDA architectures for geopotential height forecasting using NCEP reanalysis data, a standard benchmark for long-range atmospheric variability and forecast evaluation. Its temporal coverage (1948-present) and consistent data assimilation make it well-suited for training and comparing data-driven forecasting models. Our key contributions are:

1. A comprehensive benchmark comparing six attention based model architectures (Transformer, GLA, DeltaNet, Gated DeltaNet, Mamba2, and KDA) for geopotential height forecasting, including measuring forecasting skill (root mean square error), computational throughput, and memory consumption.
2. Adapting an open-source training framework, FLAME, built on Torchtitan[17] with specialized support for linear attention models, enabling reproducible research in weather forecasting.

The remainder of the paper is organized as follows. Section 2 reviews prior work in weather forecasting and efficient attention mechanisms. Section 3 describes the problem formulation, model architecture, and training objective. Section 4 details the dataset, training configurations, and evaluation metrics. Section 5 presents forecasting accuracy and computational scaling results, and Section 6 discusses the main findings and implications. We conclude in Section 7.

2 Related Work

2.1 Machine Learning for Weather Forecasting

The application of deep learning to weather forecasting has accelerated rapidly in recent years. FourCastNet [15] demonstrated that vision transformers with Fourier neural operators could achieve competitive global forecasts. A 3D Earth-specific transformer was introduced in Pangu-Weather [2] and GraphCast [9] formulated forecasting as message passing on a multi-scale mesh. GenCast [16] a diffusion-based probabilistic model is the current state-of-the-art in data-driven medium-range prediction. ClimaX [12], Stormer [13] are transformer based models used for weather forecasting research. Most of these approaches predominantly rely on transformer architectures with standard softmax attention. While effective, the quadratic complexity limits their ability to process very long temporal sequences or high-resolution spatial data without resorting to windowed

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

attention or other approximations [10] [19]. Our work explores how linear attention mechanisms can maintain forecasting accuracy while overcoming these computational limitations.

2.2 Linear Attention Mechanisms

Linear attention replaces the softmax normalization in standard attention with feature maps, enabling a recurrent formulation with linear complexity [8]. Given queries Q, keys K, and values V, standard attention computes

$$\mathrm{Attn}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V, \tag{1}$$

whereas linear attention applies feature maps $\phi(\cdot)$ to obtain

$$\mathrm{LinAttn}(Q,K,V) = \phi(Q)\big(\phi(K)^\top V\big). \tag{2}$$

State space models (SSMs) provide an alternative linear-time formulation for sequence modeling [5]. Mamba introduces selective state updates with input-dependent parameters [4], and **Mamba2** further refines this formulation by establishing a duality between selective SSMs and structured attention [3]. Under this view, the state update can be written as

$$S_t = \alpha_t S_{t-1} + \beta_t k_t v_t^\top, \tag{3}$$

where past state information is selectively decayed and updated using current inputs.

Gated Linear Attention (GLA) [22] extends linear attention by introducing fine-grained, channel-wise decay. The recurrent state update is given by

$$S_t = \mathrm{Diag}(\alpha_t)S_{t-1} + k_t v_t^\top, \tag{4}$$

DeltaNet [18] introduces error-correcting learning via the classical delta rule. By minimizing the squared prediction error, the state update becomes

$$S_t = S_{t-1} + \beta_t k_t\big(v_t - S_{t-1}^\top k_t\big)^\top, \tag{5}$$

which improves stability and long-term credit assignment.

Gated DeltaNet (GDN) [21] combines the delta rule with selective decay inspired by Mamba-style updates, extending DeltaNet with input-dependent forgetting.. Performing stochastic gradient descent on a decayed state yields

$$S_t = (I - \beta_t k_t k_t^\top)\,\alpha_t S_{t-1} + \beta_t k_t v_t^\top, \tag{6}$$

integrating error correction with input-dependent forgetting.

Kimi Delta Attention (KDA) [27] further generalizes GDN by introducing channel-wise decay within the delta-rule framework. The resulting update is

$$S_t = (I - \beta_t k_t k_t^\top)\,\mathrm{Diag}(\alpha_t)S_{t-1} + \beta_t k_t v_t^\top, \tag{7}$$

providing a unified optimization-based perspective on linear attention, selective SSMs, and delta-rule learning.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

2.3 Geopotential Height and Atmospheric Dynamics

Geopotential height measures the height of a constant-pressure surface above mean sea level, weighted by gravitational acceleration. It encodes the thermal structure of the atmosphere: warm columns expand and raise pressure surfaces, while cold columns contract and lower them. We focus on the 500 hPa level (Z500) because it is a standard benchmark for evaluating weather forecast quality. Unlike fields such as humidity or precipitation, Z500 exhibits smooth spatial gradients and is largely independent of local surface conditions such as topography, yet it retains the important global flow features—including midlatitude jets and the pole-to-equator geopotential gradient—that characterize large-scale atmospheric circulation [6,11].

3 Methods

3.1 Problem Formulation

We formulate geopotential height forecasting as a sequence-to-sequence regression problem. Given a sequence of T_{in} input frames $X = \{x_1, x_2, \ldots, x_{T_{in}}\}$ where each frame $x_t \in \mathbb{R}^{C \times H \times W}$ represents geopotential height at C pressure levels on a latitude-longitude grid of size $H \times W$, the goal is to predict the next T_{out} frames $Y = \{y_1, y_2, \ldots, y_{T_{out}}\}$.

3.2 Model Architecture

We adopt a unified architecture framework that wraps different attention backbones (Fig. 1). The architecture consists of three components:

Input Projection: Each input frame is flattened and projected to the model's hidden dimension:

$$h_t^{(0)} = W_{proj}[\text{flatten}(x_t)] + b_{proj} \tag{8}$$

where $W_{proj} \in \mathbb{R}^{d_{model} \times (C \cdot H \cdot W)}$.

Sequence Backbone: The projected sequence is processed by L layers of the attention mechanism under evaluation (Transformer, GLA, Gated DeltaNet, KDA, Mamba2, or DeltaNet):

$$h^{(l)} = \text{Layer}^{(l)}(h^{(l-1)}), \quad l = 1, \ldots, L \tag{9}$$

Forecast Head: The final hidden state is mapped to the output prediction:

$$\hat{Y} = \text{reshape}(W_{head} h_{T_{in}}^{(L)} + b_{head}) \tag{10}$$

where $W_{head} \in \mathbb{R}^{(T_{out} \cdot C \cdot H \cdot W) \times d_{model}}$.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

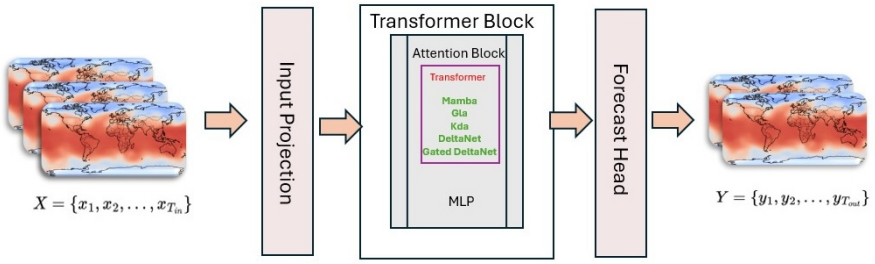


Fig. 1: Unified model architecture for geopotential height forecasting. The attention backbone is interchangeable, enabling fair comparison across mechanisms.

3.3 Training Objective

We minimize the mean squared error between predictions and ground truth:

$$\mathcal{L} = \frac{1}{T_{out} \cdot C \cdot H \cdot W} \sum_{t=1}^{T_{out}} \|y_t - \hat{y}_t\|_2^2 \tag{11}$$

Data is normalized per-level using pre-computed statistics from the training set:

$$\tilde{x}_t^{(c)} = \frac{x_t^{(c)} - \mu^{(c)}}{\sigma^{(c)}} \tag{12}$$

where $\mu^{(c)}$ and $\sigma^{(c)}$ are the mean and standard deviation for pressure level c.

4 Experimental Setup

4.1 Dataset

We use NCEP/NCAR Reanalysis 1 geopotential height data [7], which provides 6-hourly global atmospheric fields from 1948 to present. The dataset is on a 2.5×2.5 latitude–longitude grid (73×144 points) with 17 pressure levels spanning 1000–10 hPa; unless otherwise stated, we report results at 500 hPa, a standard benchmark level in meteorology. We use 6-hourly snapshots with a chronological 90%/10% train–validation split. Table 1 summarizes the normalization statistics for selected pressure levels.

Table 1: Normalization statistics for geopotential height at selected pressure levels.

Level (hPa)	Mean (m)	Std (m)	Min (m)	Max (m)
500	5,509.7	343.3	4,369.0	6,063.0

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

4.2 Model Configurations

To ensure a fair comparison, all models are configured to approximately 340M parameters (Table 2), with architecture-specific hyperparameters set following published recommendations; all configuration files are released in our GitHub repository. The transformer applies Rotary Position Embeddings (RoPE), while the linear attention variants do not use explicit positional encoding in their recurrent layers, and Mamba2 encodes ordering implicitly through its state-space recurrence. Unless otherwise noted, evaluations use a 48-hour lead time. For throughput and memory benchmarking, we report performance on a single-step output (6-hour lead). For baseline comparisons against persistence and climatology, we use a 24-hour horizon with $T_{out} = 4$ and report per-timestep errors at $t0$–$t3$ (corresponding to 6, 12, 18, and 24 hours).

Table 2: Model configurations for the architecture comparison. All models have approximately 340M parameters.

Model	Layers	Hidden	Heads	Head Dim	Params
Transformer	24	1024	16	64	∼340M
GLA	24	1024	4	256	∼340M
Gated DeltaNet	21	1024	6	256	∼340M
DeltaNet	24	1024	8	128	∼340M
Mamba2	48	1024	–	64	∼340M
KDA	21	1024	6	256	∼340M

4.3 Training Configuration

All models are trained using the FLAME framework with consistent hyperparameters as listed in (Table 3). The model implementations are available in [24]. Training runs were executed on the Polaris system at the Argonne Leadership Computing Facility using 2 nodes. Each node provides 4 NVIDIA A100 GPUs (connected via NVLink) with 40GB of HBM per GPU. We use all 8 GPUs in total for distributed training using the TorchTitan framework for each model. We flatten the spatial grid is flattened to a 10,512-dimensional feature vector (CHW) per timestep, and the data is normalized with precomputed statistics for the pressure level considered.

4.4 Evaluation Metrics

Model Performance Metrics We evaluate model training and validation using standard meteorological metrics for a given batch of data. In all our analysis we average the metrics over a consistent batch of 256 samples: Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{13}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

Table 3: Training configuration for geopotential height forecasting.

Parameter	**Value**
Pressure level	500 hPa
Input length (T_{in})	8 timesteps (48 hours)
Target length (T_{out})	1-4 timesteps (6-24 hours)
Global batch size	256
Optimizer	AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-15}$
Learning rate	3×10^{-4} with cosine decay
Warmup steps	1,024
Total training steps	5,000
Gradient clipping	Max norm 1.0
Precision	Mixed precision (bfloat16)

where y_i is the true, observed value for the i-th data point in the dataset, and $\hat{y}_i$ is the value predicted by the model corresponding to the i-th data point.

Computational Metrics: We report training throughput in tokens/second, measured as the effective processed tokens per second across all GPUs, to capture end-to-end training efficiency. Peak GPU memory usage (GB) is recorded during the forward/backward pass to quantify the memory footprint at each sequence length. Inference latency is reported in milliseconds per sample and averaged over multiple batches with synchronized timing to reflect steady-state deployment performance.

Baselines Metrics: To contextualize model performance, we include two simple baselines that bracket short-term continuity and long-term mean behavior. Persistence predicts the last input frame and tests whether a model adds skill beyond trivial temporal extrapolation ($\hat{y}_t = x_{T_{in}}$); climatology predicts the per-batch mean field and provides a no-skill reference in anomaly space ($\hat{y}_t = \bar{y}$).

5 Results

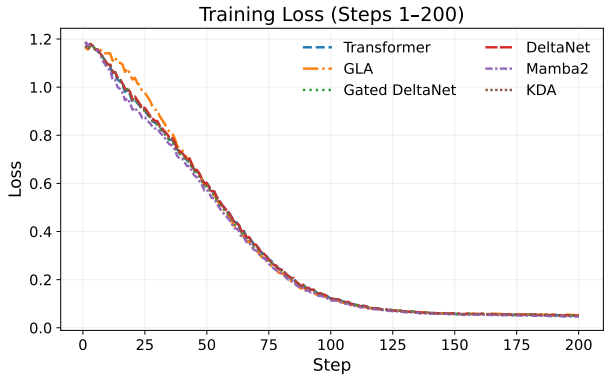
We summarize forecasting performance, training dynamics, and efficiency across attention backbones. We present the accuracy comparison at 24-hour lead time, analyze forecast-horizon behavior using per-timestep errors, and finally report throughput and memory scaling at long sequence lengths.

5.1 Model Comparison

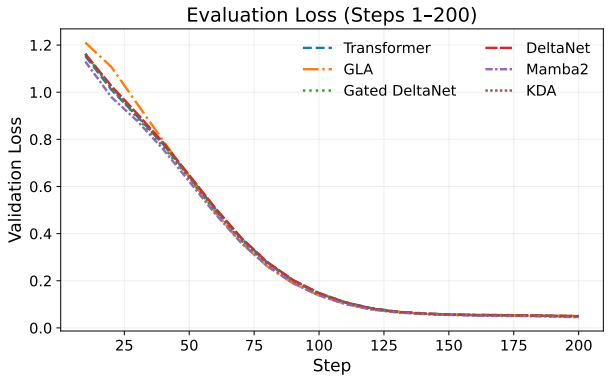
Table 4 presents the forecasting performance of all architectures at 6-hour lead time on 500 hPa geopotential height data. For model evaluation, we train each model for 5,000 steps and evaluate on the validation dataset every 20 steps. The early training curves (Fig. 2) indicate rapid convergence within the first 200–500 steps, with linear attention variants typically reaching lower evaluation loss earlier than the Transformer. By step 5,000, linear attention models achieve the strongest validation loss and ACC; GLA and DeltaNet yield the lowest validation losses (0.003550 and 0.003559) and the highest ACC values (0.920441 and

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

0.920416). Mamba2 and Gated DeltaNet remain competitive in training RMSE with slightly lower ACC. The Transformer shows earlier signs of overfitting, with its validation loss and ACC lagging behind despite continued training. Throughput and memory remain in the same operating regime across models at this step, with Gated DeltaNet and KDA exhibiting higher TFLOPS but also the largest memory footprints.



(a) Training loss (steps 1–200).



(b) Eval loss (steps 1–200).

Fig. 2: Training and evaluation loss on HGT over the first 200 steps for Transformer, GLA, DeltaNet, Gated DeltaNet, KDA, and Mamba2.

Fig. 3 and 4 show GLA forecasts at step 5000 for training and validation samples. The predicted fields capture the large-scale geopotential height patterns with coherent gradients, and the validation example exhibits similar structure to the training case, suggesting good generalization at this training stage. Other models are show very comparable forecasts at this time step.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

Table 4: Snapshot metrics at step 5000.

Model	Tr. RMSE	Val loss	Val ACC	Tokens/s	TFLOPS	Mem(GB)
Transformer	4.4e-2	6.3e-3	0.879579	31,980	95.19	4.28
GLA	4.1e-2	3.6e-3	0.920441	31,213	93.02	4.26
GDN	3.6e-2	4.2e-3	0.911715	29,313	109.32	5.91
DeltaNet	3.8e-2	3.6e-3	0.920416	31,875	94.97	4.31
Mamba2	3.6e-2	4.8e-3	0.903341	30,261	109.89	4.38
KDA	3.7e-2	3.8e-3	0.920187	26,247	98.74	5.92

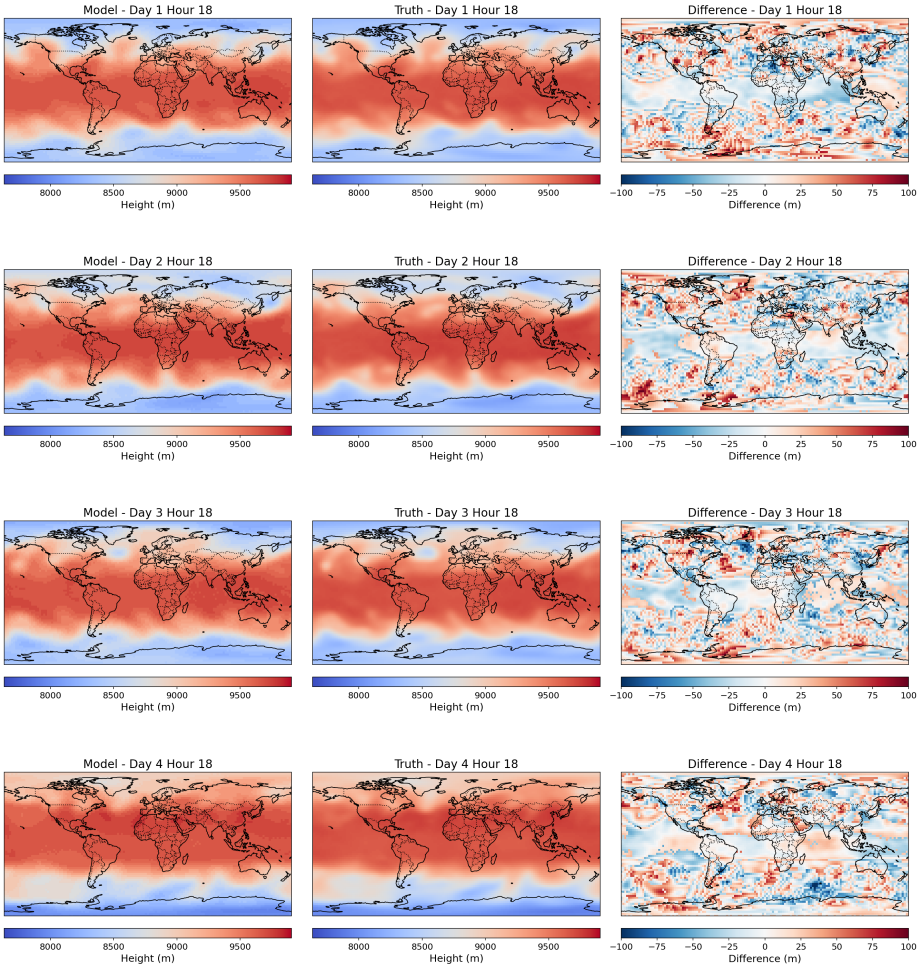


Fig. 3: Train forecast at step 5000.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

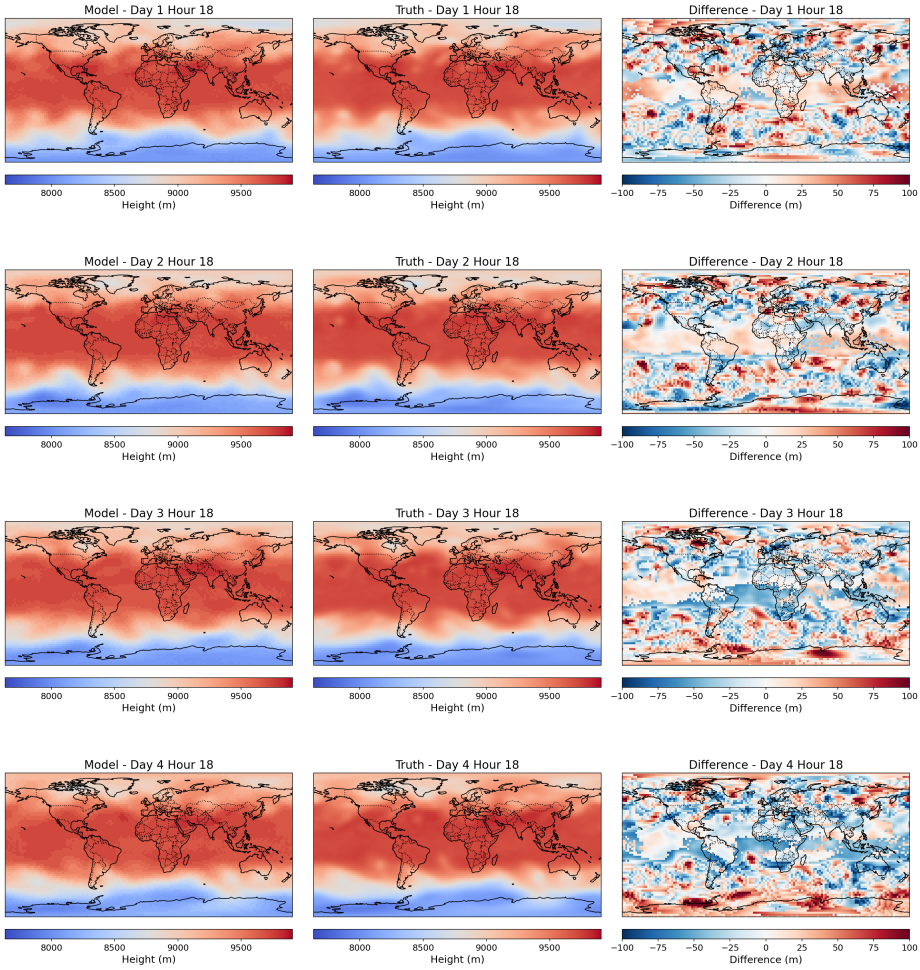


Fig. 4: Validation forecast at step 5000.

5.2 Forecast Horizon Analysis

At a 24h lead time ($T_{out} = 4$), climatology yields $ACC \approx 0$ and high $RMSE \approx$ 118m, reflecting its inability to capture day-scale variability once the mean field is removed. Persistence is substantially stronger at short horizons, achieving $RMSE \approx 63.6$m and $ACC \approx 0.826$, which is expected given the strong temporal autocorrelation of geopotential height. However, persistence errors increase rapidly with forecast horizon. Per-timestep RMSE at step 5000 shows that persistence is best at the first step ($t0 = 31.6$m) but degrades sharply by $t3 = 85.3$m. In contrast, both the full-scale attention transformer model and linear attention models (shown with an example of GLA) maintain lower errors at longer horizons, with GLA consistently outperforming transformer across all four steps (e.g., $t3$: 59.2m vs 66.6m) and exhibiting higher anomaly correlation

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

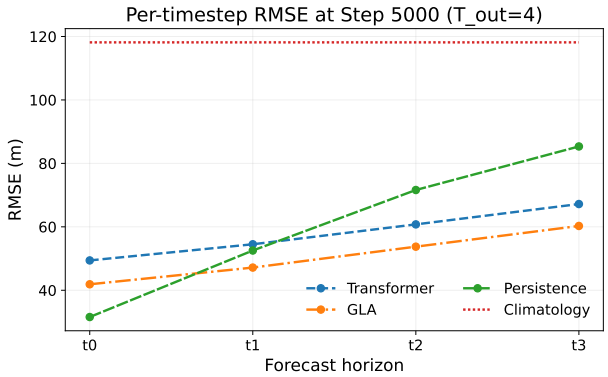


Fig. 5: Per-timestep RMSE at step 5000 for $T_{out} = 4$, comparing Transformer, GLA, persistence, and climatology.

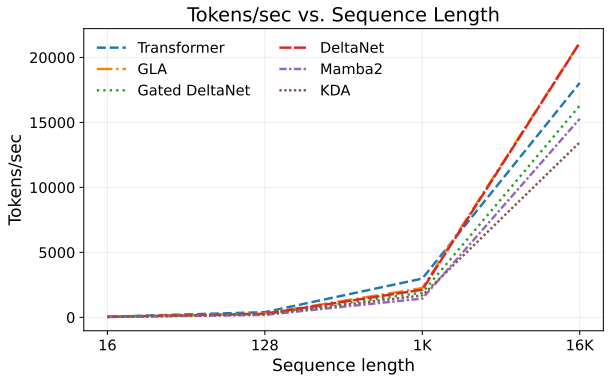
($ACC \approx 0.873$ vs 0.826). Together, these comparisons indicate that linear attention models—particularly GLA—learn temporal dynamics beyond last-frame copying, yielding improved multi-step forecasts at day-scale lead times 5.

5.3 Computational Efficiency

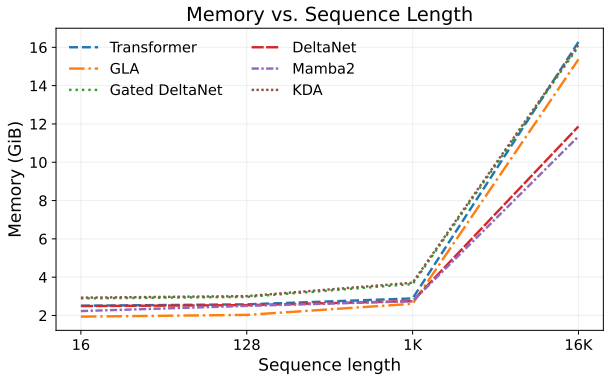
The throughput and memory scaling results as shown in Fig. 6 are measured at sequence lengths 16, 128, 1K, and 16K. Across all models, tokens/sec increases with sequence length, reflecting higher token throughput per optimization step as the sequence grows. At 16K, linear attention models such as GLA and DeltaNet show higher token throughput than the standard Transformer (21.1k vs 18.0k tokens/sec, a 17% improvement), while Mamba2 is lower (15% below Transformer). Memory trends are more discriminative: at 16K, the Transformer reaches 16.28 GiB, whereas Mamba2 uses 11.35 GiB (about 30% lower), and DeltaNet remains similarly low at 11.87 GiB. These results indicate that linear attention variants can sustain higher long-sequence throughput while also reducing peak memory at large context lengths.

At an extended context length of 131K with tensor parallelism = 16, GLA achieves higher throughput (2,030 vs 1,906 tokens/sec) than the full-attention Transformer, with a modest increase in memory footprint (32.09 GiB vs 26.25 GiB), indicating that GLA sustains better long-context throughput. The higher memory usage stems from GLA's gated linear attention design, which adds extra projection and gating paths (e.g., additional $q/k/v/g$ projections and gating normalization), increasing activation and intermediate buffer memory. At moderate lengths, the Transformer's quadratic attention activations dominate total memory, so the extra GLA activations are comparatively less significant. At extremely long sequences and high tensor parallel degree, however, those additional gating buffers—combined with a less-optimized TP implementation for

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13



(a) Tokens/sec vs. sequence length.



(b) Memory vs. sequence length.

Fig. 6: Throughput and memory scaling at sequence lengths 16, 128, 1K and 16K.

GLA—can outweigh the Transformer's advantage, leading to higher measured memory. We expect this balance to shift with further sequence-length increases or with improved TP/fusion optimizations for GLA, revealing stronger memory advantages for linear attention at extreme contexts.

6 Discussion

Our results show that linear attention backbones can achieve competitive forecasting accuracy while scaling more effectively to long temporal contexts. The persistence baseline remains strong at short horizons, reflecting the inherent autocorrelation in geopotential height; nonetheless, linear attention models show clearer advantages as lead time increases, indicating learned dynamics beyond last-frame extrapolation. Among the linear variants, GLA consistently attains strong validation metrics, suggesting that gating and data-dependent decay help

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

preserve useful long-range information. The scaling experiments reinforce that quadratic attention is the dominant bottleneck at long sequence lengths, whereas linear attention maintains higher throughput with a more gradual memory increase. At extreme contexts, some linear attention models still exhibit higher memory usage, which likely stems from model-parallel implementations that are not yet fully optimized. Overall, these trends support linear attention as a practical and scalable alternative to standard transformers for long-horizon atmospheric forecasting.

7 Conclusion

We presented a systematic comparison of linear attention architectures for geopotential height forecasting and bench-marked them against standard transformers on NCEP reanalysis data. The results indicate that linear attention models can match or improve forecasting quality while offering improved scalability at long sequence lengths, with GLA providing consistently strong validation performance. The FLAME framework enables reproducible training and evaluation of these models with distributed data parallelism and is released as open source to facilitate further research.

Limitations and Future Work: Our study focuses on single-variable forecasting at fixed spatial resolution and evaluates a limited set of lead times. Future work will extend to multivariate prediction, higher-resolution data, longer forecast horizons, and optimized parallel implementations for linear attention models.

8 Acknowledgment

This research used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357 and the work is supported by the Office of Science, U.S. Department of Energy, under contract DE-AC02-06CH11357.

References

1. Bauer, P., Thorpe, A., Brunet, G.: The quiet revolution of numerical weather prediction. Nature **525**(7567), 47–55 (2015)
2. Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., Tian, Q.: Accurate medium-range global weather forecasting with 3D neural networks. Nature **619**(7970) (2023)
3. Dao, T., Gu, A.: Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality (2024)
4. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2024)
5. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. In: International Conference on Learning Representations (2022)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

6. Holton, J.R., Hakim, G.J.: An Introduction to Dynamic Meteorology. Academic Press, 5th edn. (2013)
7. Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., et al.: The NCEP/NCAR 40-year reanalysis project. Bulletin of the American Meteorological Society **77**(3), 437–472 (1996)
8. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are RNNs: Fast autoregressive transformers with linear attention. In: ICML. PMLR (2020)
9. Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., et al.: Learning skillful medium-range global weather forecasting. Science **382**(6677), 1416–1421 (2023)
10. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
11. Maulik, R., Rao, V., Wang, J., Mengaldo, G., Constantinescu, E., Lusch, B., Balaprakash, P., Foster, I., Kotamarthi, R.: Efficient high-dimensional variational data assimilation with machine-learned reduced-order models. Geoscientific Model Development **15**(8), 3433–3445 (2022)
12. Nguyen, T., Brandstetter, J., Kapoor, A., Gupta, J.K., Grover, A.: ClimaX: A foundation model for weather and climate. In: ICML. PMLR (2023)
13. Nguyen, T., Shah, R., Bansal, H., Arcomano, T., Maulik, R., Kotamarthi, V., Foster, I., Madireddy, S., Grover, A.: Scaling transformer neural networks for skillful and reliable medium-range weather forecasting (2024), `https://arxiv.org/abs/2312.03876`
14. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems **32** (2019)
15. Pathak, J., et al.: FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. arXiv preprint arXiv:2202.11214 (2022)
16. Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T.R., El-Kadi, A., Masters, D., Ewalds, T., Stott, J., Mohamed, S., Battaglia, P., Lam, R., Willson, M.: Gencast: Diffusion-based ensemble forecasting for medium-range weather (2024), `https://arxiv.org/abs/2312.15796`
17. PyTorch Team: TorchTitan: A native PyTorch library for large model training. `https://github.com/pytorch/torchtitan` (2024)
18. Schlag, I., Irie, K., Schmidhuber, J.: Linear transformers are secretly fast weight programmers. In: ICML. pp. 9355–9366. PMLR (2021)
19. V., H., et al.: Aeris: Argonne earth systems model for reliable and skillful predictions (2025), `https://arxiv.org/abs/2509.13523`
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
21. Yang, S., Kautz, J., Hatamizadeh, A.: Gated delta networks: Improving mamba2 with delta rule. In: Proceedings of ICLR (2025)
22. Yang, S., Wang, B., Shen, Y., Panda, R., Kim, Y.: Gated linear attention transformers with hardware-efficient training. In: Proceedings of ICML (2024)
23. Yang, S., Wang, B., Zhang, Y., Shen, Y., Kim, Y.: Parallelizing linear transformers with the delta rule over sequence length. In: Proceedings of NeurIPS (2024)
24. Yang, S., Zhang, Y.: Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism (Jan 2024), `https://github.com/fla-org/flash-linear-attention`

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13

25. Zhang, Y., Yang, S.: Flame: Flash language modeling made easy (Jan 2025), https://github.com/fla-org/flame
26. Zhang, Y., Yang, S., Zhu, R., Zhang, Y., Cui, L., Wang, Y., Wang, B., Shi, F., Wang, B., Bi, W., Zhou, P., Fu, G.: Gated slot attention for efficient linear-time sequence modeling. In: Proceedings of NeurIPS (2024)
27. Zhang, Y., et al.: Kimi linear: An expressive, efficient attention architecture (2025)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_13