GraphMet: Learning Spectral Compositional Koopman Operators for Global-to-Regional Weather Forecasting

Khalid Oublal[*1,2,6] **Malo Guichard**[2,‡] **François Bertholom**[1,‡] **Simon Albergel**[2]
David Benhaiem[2] **Emmanuel Le Borgne**[2] **Vladimir R. Kostic**[3,5] **Karim Lounici**[1,6]

[1]École Polytechnique, Institut Polytechnique de Paris, France [2]OneTech, TotalEnergies, France
[3]CSML, Istituto Italiano di Tecnologia, Genova, Italy [5] University of Novi Sad, Serbia
[6]CMAP, École Polytechnique, Palaiseau, France

Abstract

State-of-the-art machine learning weather prediction systems achieve strong upper-air forecast skill but degrade at longer lead times, with mesoscale structures progressively vanishing and ensemble forecasts becoming under-dispersive. While these limitations are often attributed to the training objective, we find that message-passing processors significantly contribute to this degradation. Repeated neighborhood aggregation at each autoregressive step acts as a smoothing operator that, over rollout steps, suppresses high-frequency information in latent space. This effect is particularly limiting in regional weather prediction, where fine-scale structures are already partially unresolved, further eroding mesoscale variability and hindering long-horizon forecasting. We propose **GraphMet**, a **Graph**-Based **Met**eorological forecasting model inspired by Koopman theory, which replaces stacked message-passing layers with a single linear Koopman operator step. GraphMet learns linear evolution in a spectral latent basis while preserving the encoder, decoder, and multi-scale icosahedral mesh structure. A two-term spectral objective jointly learns a stable Koopman basis and preserves latent geometry, enabling efficient long-range propagation without repeated nonlinear aggregation. Empirically, GraphMet reduces the 15-day activity deficit by 32% and improves regional RMSE by 11–14% at 5 km resolution, and enables efficient probabilistic forecasting with lower inference cost than strong baselines. Code and leaderboard available at 🤗 https://huggingface.co/koublal/GraphMet.

1 Introduction

Machine Learning Weather Prediction (MLWP) is a fast-evolving field, in which current state-of-the-art methods use deep neural architectures trained on large-scale reanalysis data. GraphCast [17], AIFS [18], Pangu-Weather [2], and GenCast [28] all rely on an encode-process-decode structure to exploit the powerful modeling capabilities of Graph Neural Networks (GNNs) for handling irregular meshes and multi-scale spatial dynamics. Atmospheric variables are lifted from a latitude-longitude grid onto a mesh, processed by a GNN, and projected back onto the original grid. While these methods exhibit strong performance for short-to-medium term forecasting, they struggle to model high-frequency spatial structures over longer time horizons, leading to over-smoothing and degraded prediction quality. This is particularly detrimental in settings where it is essential to model fine-grained dynamics correctly, e.g., regional forecasting.

*Corresponding author: khalid.oublal@polytechnique.edu. Code: https://github.com/koublal/GraphMet, Notebook and Leaderboard 🤗 https://huggingface.co/koublal/GraphMet. ‡ Equal contributions.

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

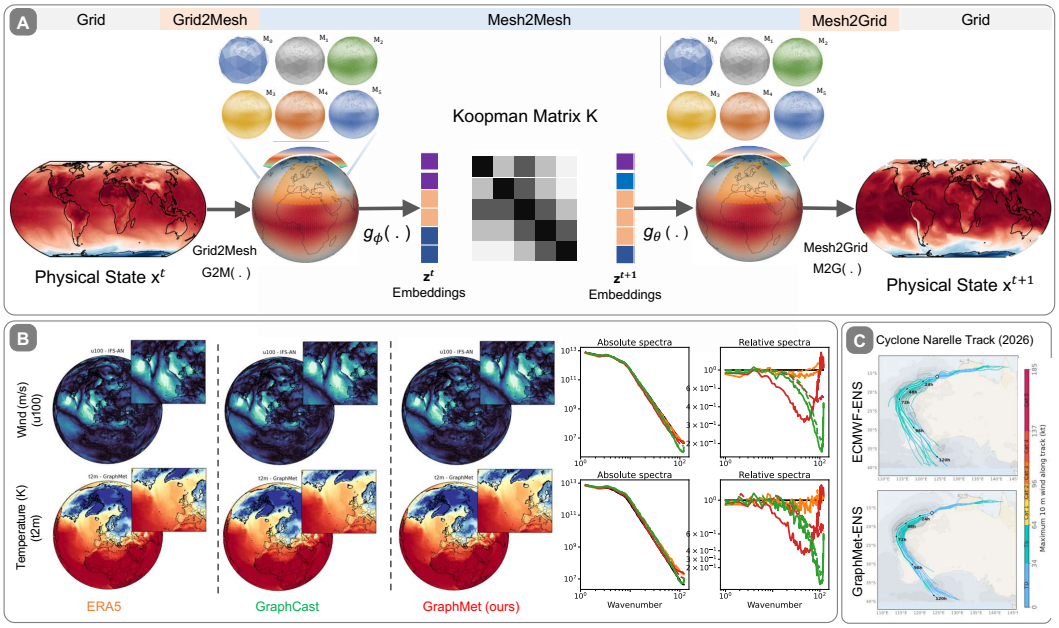


Figure 1: Overview of GraphMet. **(A)** Global forecasting from state $\mathbf{x}^t$ using a multiscale mesh encoder (levels 0–6), latent Koopman dynamics, and a decoder for next-state prediction. **(B)** Prediction example comparing oversmoothing in GraphCast and GraphMet against ERA5, supported by the spectral analysis on the right. **(C)** ECMWF-ENS and GraphMet-ENS forecasts for Tropical Cyclone Narelle (Australia, March 2026), showing 50-member ensemble tracks with the physical ECMWF IFS-ENS in top.

This over-smoothing phenomenon is commonly attributed to the Mean-Squared Error (MSE) used in training objectives. Prior works link this to reduced forecast activity and ensemble averaging effects [17, 18]. However, recent probabilistic models like GenCast [28], GraphEFM [26], and DiffLAM [20] for regional (i.e., limited-area modeling, LAM) weather forecasting still exhibit the same issue, suggesting that the fundamental bottleneck is not the objective function. In parallel, the GNN community has observed and extensively studied the over-smoothing problem inherent to deep message-passing architectures [21, 24, 4]. We hypothesize that GNN processors are the main culprit for eliminating high-frequency structures in long-range forecasting, and study the problem through the lens of Koopman operator theory. We propose the novel GraphMet method, that builds upon the general encode-process-decode pipeline but replaces the complex processor by a linear approximator, trained via a principled objective. Our approach is motivated by prior works establishing that atmospheric dynamics admit a spectral decomposition in which evolution is approximately linear in the leading eigenspace [32, 23, 14].

Contributions. Our main contributions are as follows:

- (i) We show that spatial over-smoothing observed in prior models is a consequence of architectural choices, and interpret a GNN over-smoothing result in the Koopman approximation setting. We derive a simpler and more effective architecture from these observations.
- (ii) We introduce a new loss function to learn approximate Koopman operators that provably recovers the optimal singular functions of the true operator in the Eckart-Young-Mirsky sense, for any compact operator and with no normality or self-adjointness assumption on the atmospheric dynamics.
- (iii) The resulting model yields a 32% activity-deficit reduction for week-long forecasts, an improvement in the regional Root-Mean-Squared Error (RMSE) of 11 to 14% at the 5.5 km scale, and competitive CRPS at *fourteen times lower inference cost* compared to the strongest baselines available.

Outline We start by introducing the general setting and Koopman formalism in Section 2. Then, we exhibit the structural limits of existing methods for predicting high-frequency spatial dynamics in

medium-long range forecasts (Section 3). We describe our method and provide theoretical grounding in Section 4. Related works are discussed in Section 5 and we provide extensive empirical validation for our method in Section 6.

2 Modeling Atmospheric Dynamics with Koopman Operators

We start by defining the coordinate system in which MLWP models operate, which naturally leads us to formalizing a framework based on Koopman operator theory to model atmospheric dynamics. The spectral decomposition of the Koopman operator allows us to cleanly and intuitively separate long-lived, global trends from short-lived, regional details.

Formalism. We let $\mathbf{x}_t \in \mathcal{X}$ denote the atmospheric state at time t, where $\mathcal{X} \subseteq \mathbb{R}^{N_G \times d}$. Here, N_G denotes the number of points in a predefined latitude-longitude grid, and d is the number of prognostic fields, i.e., physical variables we want to predict (e.g., temperature, humidity, wind speed, etc.). Although conceptually intuitive, learning dynamics on latitude-longitude grids makes simulating global dynamics inefficient because of projection errors. Models like GraphCast [17] solve this by mapping grid data onto an icosahedral mesh with N_M nodes. This gives a representation in the space $\mathcal{Y} = \mathbb{R}^{N_M \times d}$. During the learning process, the data is further encoded so that each node in the mesh holds an m-dimensional embedding of the local weather. Hence the atmospheric dynamics are effectively learned in the latent space $\mathcal{Z} = \mathbb{R}^{N_M \times m}$.

In summary, the model consists of grid-to mesh and mesh-to grid operators $\mathrm{G2M}$ and $\mathrm{M2G}$, along with a time-independent encoder $g_\eta : \mathcal{Y} \to \mathcal{Z}$ and a decoder $g_\psi : \mathcal{Z} \to \mathcal{Y}$. The full encoder and decoder are

$$\boldsymbol{f_\phi} = g_\eta \circ \mathrm{G2M}, \qquad \boldsymbol{g_\theta} = \mathrm{M2G} \circ g_\psi. \tag{1}$$

The pipeline is visualized in Figure 1-B.

Koopman Operator. Ideally, we would like to learn a universal operator capable of predicting the weather independently of time and space. This means we want to model atmospheric states as a stationary ergodic Markov process on the measure space $(\mathcal{Z}, \Sigma, \mu)$, where Σ is the Borel σ-field of $\mathcal{Z}$ and μ is a σ-finite measure. Building such a model using hand-picked physical variables appears as a complex and unreliable method. Instead of tracking non-linear physical states directly, we rely on Koopman operator theory and work with observables, i.e., real-valued functions in $L^2(\mathcal{Z}, \mu)$. The Koopman operator $\mathcal{K} : L^2(\mathcal{Z}, \mu) \to L^2(\mathcal{Z}, \mu)$ is defined for $h \in L^2(\mathcal{Z}, \mu)$ by

$$[\mathcal{K}h](\mathbf{z}) = \mathbb{E}[h(\mathbf{z}^{t+1}) \mid \mathbf{z}^t = \mathbf{z}], \quad \text{for all } \mathbf{z} \in \mathcal{Z}. \tag{2}$$

To derive a physically meaningful Singular Value Decomposition (SVD) of the Koopman operator, we assume that $\mathcal{K}$ is a Hilbert-Schmidt operator.

Assumption 2.1. Let $\{h_i : i \geq 1\} \subset L^2(\mathcal{Z}, \mu)$ be a countable orthonormal basis of $L^2(\mathcal{Z}, \mu)$. We let $\|\mathcal{K}\|_{\mathrm{HS}}$ denote the Hilbert-Schmidt norm of the Koopman operator $\mathcal{K}$, defined by

$$\|\mathcal{K}\|_{\mathrm{HS}}^2 = \textstyle\sum_{i \geq 1} \|\mathcal{K}h_i\|_{L^2(\mu)}^2, \tag{3}$$

where $\|h\|_{L^2(\mu)} = \int |h|^2 \mathrm{d}\mu$ for all $h \in L^2(\mathcal{Z}, \mu)$. We assume that the Koopman operator $\mathcal{K}$ is a Hilbert-Schmidt operator, i.e., $\|\mathcal{K}\|_{\mathrm{HS}} < \infty$.

Under Assumption 2.1, the operator $\mathcal{K}$ admits an singular value decomposition (SVD) and, under stronger generic assumptions, an eigenvalue decomposition (EVD) [6]. That is, we can write

$$\mathcal{K} = \textstyle\sum_{i \geq 1} \sigma_i \, u_i \otimes v_i = \sum_{i \geq 1} \lambda_i \, \varphi_i \otimes \psi_i, \tag{4}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots > 0$ are the singular values, the u_i's and v_i's are left and right singular functions in $L^2(\mathcal{Z}, \mu)$, and $\otimes$ denotes the outer product in $L^2(\mathcal{Z}, \mu)$. On the other hand, $(\lambda_i) \in \mathbb{C}^{\mathbb{N}}$ are eigenvalues with left and right eigen-functions $\varphi_i, \psi_i \in L^2(\mathcal{Z}, \mu)$, respectively. The SVD exposes the effective dimension of the dynamics through the gap between consecutive singular values $\sigma_r > \sigma_{r+1}$, providing an optimal rank-r approximation of $\mathcal{K}$ in basis of $(\varphi_i)_{i=1}^r$ via the Eckart-Young theorem. If the operator $\mathcal{K}$ is normal, the singular- and eigen-functions coincide. Moreover, the EVD of an r-truncated SVD of $\mathcal{K}$ is generally a good approximation of the full EVD of $\mathcal{K}$. The EVD has the advantage of decomposing atmospheric dynamics into interpretable, independent components. For $i \geq 1$, the eigenvalue λ_i characterizes the lifespan of some weather pattern. Large $|\lambda_i|$'s correspond

to long-lived modes that last for several weeks (e.g., planetary Rossby waves, that dictate global trends), while small $|\lambda_i|$'s represent short-lived modes that are typically dissipated in about a day and have local consequences. The sensing function ψ_i describes how a pattern influences the weather, while φ_i encodes its physical structure. Preserving short-lived dynamics is essential for accurate regional forecasts, as smoothing models tend to suppress local variability and retain only large-scale behaviour. In the next section, we show that existing Graph Neural Network (GNN) approaches struggle to preserve these dynamics over long time horizons.

3 Structural Limits of GNN-Based Weather Processors

When modeling atmospheric dynamics in practice, we must approximate the infinite-dimensional Koopman operator $\mathcal{K}$ using a finite-dimensional projection. Ideally, such a projection into a latent space of dimension r should amount to truncating the SVD of $\mathcal{K}$ to its first r terms, ranked by scale. However, an arbitrary encoder $\boldsymbol{f}_{\boldsymbol{\phi}}$ has no reason to inherently span this optimal subspace.

In this section, we analyze why existing architectures based on Graph Neural Networks (GNNs) fail to capture this projection over extended rollouts. We start by establishing an exact decomposition of the approximation error made by the learned projection. Then, we prove that the oversmoothing problem of GNNs eliminates the high-frequency spatial structure required for accurate regional forecasting.

3.1 Latent Geometric Distortion

We start this section with some definitions. For convenience, we assume that the encoder $\boldsymbol{f}_{\boldsymbol{\phi}}$ in (1) is from $\mathcal{X}$ to $\mathbb{R}^r$, where $r = N_M \times m$. This simply amounts to flattening the space. We consider the finite-dimensional subspace $V_{\boldsymbol{\phi}} \subset L^2(\mathcal{Z}, \mu)$ spanned by the components of $\boldsymbol{f}_{\boldsymbol{\phi}}$, and the orthogonal projector $\mathcal{P}_\phi$ onto V_ϕ. We also define the *latent covariance matrix* $\mathbf{C}_{\boldsymbol{\phi}}$, given by

$$\mathbf{C}_{\boldsymbol{\phi}} = \int \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}) \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x})^\top \mu(\mathrm{d}\mathbf{x}), \tag{5}$$

where $\boldsymbol{f}_{\boldsymbol{\phi}}$ is the encoder in (1). We let $\lambda_{\max}(\mathbf{C}_{\boldsymbol{\phi}})$ and $\lambda^+_{\min}(\mathbf{C}_{\boldsymbol{\phi}})$ denote respectively the largest and the smallest positive eigenvalues of $\mathbf{C}_{\boldsymbol{\phi}}$. If the condition number $\kappa(\mathbf{C}_{\boldsymbol{\phi}}) = \lambda_{\max}(\mathbf{C}_{\boldsymbol{\phi}})/\lambda^+_{\min}(\mathbf{C}_{\boldsymbol{\phi}})$ is large, then the latent representation collapses along certain directions, precisely those corresponding to short-lived, mesoscale modes. When the processor averages out spatial variations, the encoder's outputs concentrate in a low-dimensional subspace, driving $\lambda^+_{\min}(\mathbf{C}_{\boldsymbol{\phi}})$ towards zero. To evaluate the quality of the learned projection, we consider the Koopman approximation error, $\|\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|^2_{\mathrm{HS}}$. It measures the information loss when the true operator is forced to evolve strictly within the finite bounds of the encoder's subspace $V_{\boldsymbol{\phi}}$.

Proposition 3.1 (Exact operator-approximation decomposition)**.** *Using the notation introduced above, let $\mathcal{P}^\perp_{\boldsymbol{\phi}} = \mathcal{I} - \mathcal{P}_{\boldsymbol{\phi}}$, where $\mathcal{I}$ denotes the identity operator on $L^2(\mathcal{Z}, \mu)$. Then the two components of the approximation error are orthogonal in the Hilbert–Schmidt inner product, $\langle \mathcal{P}^\perp_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}, \mathcal{K}\mathcal{P}^\perp_{\boldsymbol{\phi}}\rangle_{\mathrm{HS}} = 0$, and consequently*

$$\|\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|^2_{\mathrm{HS}} \;=\; \|\mathcal{P}^\perp_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|^2_{\mathrm{HS}} \;+\; \|\mathcal{K}\mathcal{P}^\perp_{\boldsymbol{\phi}}\|^2_{\mathrm{HS}} \;=\; \|\mathcal{K}\|^2_{\mathrm{HS}} - \|\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|^2_{\mathrm{HS}}\,. \tag{6}$$

In particular, since $\|\mathcal{K}\|_{\mathrm{HS}}$ does not depend on the encoder, minimizing the operator-approximation error over $\boldsymbol{\phi}$ is exactly *equivalent to maximizing the captured energy $\|\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|^2_{\mathrm{HS}}$.*

Here $\mathcal{P}^\perp_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}$ is the *leakage* of modes representable at t but not at $t+1$, and $\mathcal{K}\mathcal{P}^\perp_{\boldsymbol{\phi}}$ accounts for dynamics starting outside $V_{\boldsymbol{\phi}}$. Both components contribute to the error, and keeping them together makes the statement an identity: it involves no auxiliary norm, no spectral constant and no normality hypothesis (Section A.3).

Normalization and covariance geometry. The encoder is phase-conditioned, $\mathbf{z}_\tau = \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}, \tau)$, and $\mathbf{C}_{\boldsymbol{\phi}}(\tau)$ is exactly the Gram matrix converting Euclidean coefficient norms into ambient $L^2(\mu_\tau)$ norms, so that $\|\mathcal{P}_\tau\mathcal{K}_\tau\mathcal{P}_{\tau+\Delta}\|^2_{\mathrm{HS}} = \|\overline{\mathbf{K}}_{\tau,\Delta}\|^2_F$ for the whitened operator $\overline{\mathbf{K}}_{\tau,\Delta} = \mathbf{C}_{\boldsymbol{\phi}}(\tau)^{-1/2}\mathbf{C}_{\boldsymbol{\phi}}(\tau, \tau+\Delta)\mathbf{C}_{\boldsymbol{\phi}}(\tau+\Delta)^{-1/2}$, which is exactly invariant under $\boldsymbol{f}_{\boldsymbol{\phi}} \mapsto \lambda \boldsymbol{f}_{\boldsymbol{\phi}}$: no latent normalization is assumed and no singular value can be inflated by rescaling the encoder, which is

also why (11) is a ratio. Sections A.4 and A.5 give the derivation, the role of $\kappa(\mathbf{C}_{\boldsymbol{\phi}})$, and why no Mori–Zwanzig memory kernel arises.

Notice that the extremal eigenvalues of the latent covariance matrix $\mathbf{C}_{\phi}$ control the captured energy of Proposition 3.1, and poor conditioning may lead to latent collapse. We show that GNN-based methods tend to produce increasingly ill-conditioned covariances as message-passing depth increases. Our method (Section 4) mitigates this issue.

3.2 Oversmoothing in Graph Neural Networks and Consequences for Weather Prediction

Existing methods like GraphCast [17] process the mesh embedded features (Mesh2Mesh in Figure 1) using a GNN [34]. We let $\mathbf{H}^{(\ell)} \in \mathcal{Z}$ denote the node feature matrix at the message-passing layer ℓ. The forward pass of a standard message-passing GNN layer is defined as

$$\mathbf{H}^{(\ell+1)} = \gamma\big(\mathbf{A}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}\big), \tag{7}$$

where $\mathbf{A} \in \mathbb{R}^{N_M \times N_M}$ is the normalized adjacency matrix, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{m \times m}$ is a learnable weight matrix, and γ is a non-linear activation function applied element-wise. The Dirichlet energy of a given layer configuration $\mathbf{H} \in \mathcal{Z}$ is defined by

$$E(\mathbf{H}) = \operatorname{tr}(\mathbf{H}^{\top}(\mathbf{I} - \mathbf{A})\mathbf{H}). \tag{8}$$

Note that, by definition, the matrix $\mathbf{A}$ is row-stochastic (its entries are non-negative and each of its rows sums to 1) and symmetric. The following theorem provides a result characterizing the well-studied oversmoothing behavior of GNNs [4, 24]. The proof is given in Section A.1.

Theorem 3.2. *Let the processor be a GNN defined as in (7), and let $\lambda_2 < 1$ denote the second largest eigenvalue of the normalized adjacency matrix $\mathbf{A}$. Assume the activation function γ is s-Lipschitz continuous, and $\|\mathbf{W}^{(\ell)}\|_{\mathrm{op}} \leq 1$. After L layers, the Dirichlet energy, defined in (8), satisfies*

$$E(\mathbf{H}^{(L)}) \leq \kappa(\mathbf{I} - \mathbf{A})\,(s\lambda_2)^{2L}\,E(\mathbf{H}^{(0)}),$$

where $\kappa(\mathbf{I} - \mathbf{A})$ is the condition number of the Laplacian $\mathbf{I} - \mathbf{A}$.

Having established the exponential decay of the Dirichlet energy in GNNs, we must now examine the implications of this architectural flaw in the case of weather forecasting within our Koopman operator formalism. The following corollary is proved in Section A.2

Corollary 3.3. *Assume the setting of Theorem 3.2 and $s\lambda_2 < 1$. As L goes to infinity, the smallest positive eigenvalue of the latent covariance matrix, $\mathbf{C}_{\boldsymbol{\phi}}^{(L)} = \mathbb{E}[\mathbf{H}^{(L)}(\mathbf{H}^{(L)})^{\top}]$, denoted by $\lambda_{\min}^{+}(\mathbf{C}_{\boldsymbol{\phi}}^{(L)})$, tends to zero.*

Corollary 3.3 is formulated for a single forward pass of L message-passing layers. In practice, models operate autoregressively: the latent state $\mathbf{H}^{(L),t}$ is decoded to the grid and re-encoded to the mesh: $\mathbf{H}^{(0),t+1} = \boldsymbol{f}_{\boldsymbol{\phi}} \circ \boldsymbol{g}_{\boldsymbol{\theta}}(\mathbf{H}^{(L),t})$. Assuming $\boldsymbol{f}_{\boldsymbol{\phi}} \circ \boldsymbol{g}_{\boldsymbol{\theta}}$ does not synthesize new high-frequency spatial gradients (i.e., it is s'-Lipschitz continuous and maps any state with zero Dirichlet energy to another zero-energy state), the inter-step orthogonal residual satisfies $\|\overline{\mathbf{H}}_{\perp}^{(0),t+1}\|_F \leq s'\|\overline{\mathbf{H}}_{\perp}^{(L),t}\|_F$. Over T steps, the Dirichlet energy therefore decays as $\mathcal{O}((s'(s\lambda_2)^{2L})^T)$. Provided $s'(s\lambda_2)^{2L} < 1$, the latent covariance collapses as $T \to \infty$.

Numerical Example. For GraphCast, $\lambda_2 = 0.92$ and $s = s' = 1$, giving a contraction factor 0.92^{2L} per autoregressive step. With $L = 16$, one 6-hour step reduces the Dirichlet energy by $0.92^{32} \approx 0.069$. After one day ($T = 4$), the remaining high-frequency energy is $0.069^4 \approx 2.2 \times 10^{-5}$, and over 10 days ($T = 40$) it reaches $0.92^{1280} \approx 10^{-45}$. Even a shallow 4-layer GNN ($L = 4$) still yields $0.92^{320} \approx 10^{-12}$ over the same horizon.

Theorem 3.2 isolates the mechanism on the vanilla layer (7). MLPs, normalization and edge features enter the bound only through s and the effective aggregation operator, and for attention-based processors the same decomposition applies to the row-stochastic attention matrices; residual connections are the one practical component that weakens the worst-case bound, since the identity

Table 1: Impact of processor depth on latent conditioning and forecast skill. Deep message passing causes latent collapse ($\lambda^+_{\min}$), while the Koopman formulation preserves spectral structure and improves RMSE across variables and lead times (mean $\pm$ std, 2022 test set).

Variant	RMSE (↓) by variable and lead time										
	$\lambda^+_{\min}$ (↑)	z500		u850		t850		q700		v10	
		72h	120h	72h	120h	72h	120h	72h	120h	72h	120h
Shallow MP	4.2×10^{-2}	298±25	651±47	3.61±0.29	6.38±0.44	2.03±0.16	3.74±0.26	4.53±0.38	8.11±0.57	2.76±0.22	4.89±0.36
Deep MP	9.8×10^{-3}	312±28	687±51	3.82±0.31	6.71±0.48	2.14±0.18	3.97±0.29	4.81±0.42	8.63±0.61	2.91±0.24	5.17±0.39
Koopman (ours)	**0.83**	**248±19**	**541±37**	**3.12±0.24**	**5.63±0.37**	**1.71±0.13**	**3.14±0.21**	**3.78±0.29**	**6.79±0.45**	**2.33±0.17**	**4.14±0.29**

path is not contractive on $V_\perp$. Measured on trained checkpoints with residuals, LayerNorm and multi-scale edges in place, the per-layer Dirichlet-energy ratio has geometric mean 0.85 (GraphCast), 0.87 (GraphEFM) and 0.91 (AIFS) against the vanilla rate 0.846, versus 0.982 for one Koopman step: over-smoothing is attenuated but not removed, so the analysis identifies the mechanism without being tight (Section D.2).

Empirical Confirmation. We quantify latent collapse via $\lambda^+_{\min}(\mathbf{C}_\phi)$, with full results reported in Table 1. The main comparison is between GraphMet-deepProc (16-layer GNN processor) and GraphMet (full model), which share the same encoder, decoder, and spectral objective but differ in processor design. The deep processor shows strong collapse (9.8×10^{-3}), whereas GraphMet remains well-conditioned (0.83). The table also shows that varying the training objective (MSE vs. CRPS) leads to comparatively smaller changes in $\lambda^+_{\min}$ (7.4×10^{-3} vs. 1.1×10^{-2}), suggesting that both objectives behave similarly in this setting. Overall, GraphMet achieves an improvement of roughly two orders of magnitude in $\lambda^+_{\min}$ compared to the alternatives.

4 GraphMet: Learning Spectral Compositional Koopman Operators

4.1 Model Pipeline and Block-Sparse Koopman Processor

GraphMet uses a similar pipeline to GraphCast, see Figure 1-A. However, while GraphCast uses a 16-layer message-passing processor at the Mesh2Mesh step, GraphMet approximates the infinite-dimensional Koopman operator by a block-sparse matrix that we refer to as the Koopman matrix $\mathbf{K} \in \mathbb{R}^{r \times r}$. Recall that $r = N_M \times m$, where N_M counts the mesh nodes and m is the size of the mesh embedding. Formally, we identify the space $\mathcal{Z}$ with $\mathbb{R}^r$. At any time t and for any state $\mathbf{z}^t \in \mathcal{Z}$,

$$\mathbf{z}^{t+1} = \mathbf{K}\mathbf{z}^t, \tag{9}$$

Learning a dense Koopman matrix $\mathbf{K}$ would be infeasible in practice, since $r^2 \approx 1\text{e}12$. Thus, we adopt a block-sparse structure as in [22]. We decompose latent states node-wise, $\mathbf{z}^t = (\mathbf{z}^t_1, \ldots, \mathbf{z}^t_{N_M})$, where $\mathbf{z}^t_i \in \mathbb{R}^m$ for $i = 1, \ldots, N_M$. The block-sparse Koopman matrix acts on the i-th node only through the associated edges:

$$\mathbf{z}^{t+1}_i = \sum_{j:(j,i)\in\mathcal{E}_M} \mathbf{K}_{\alpha_{ij}} \mathbf{z}^t_j,$$

where $\mathcal{E}_M$ is the set of mesh edges, $\alpha_{ij} \in \{1, \ldots, k\}$ specifies the interaction type for the edge (j, i) among k different types, and $\mathbf{K}_{\alpha_{ij}}$ is the corresponding shared $m \times m$ sub-matrix. We use $k = 14$ interaction types (12 directional icosahedral edges plus land/ocean self-interactions). With $m = 32$, the Koopman matrix has $14 \times 32^2 = 14{,}336$ parameters. The multi-mesh structure enables multi-scale dynamics. Further architectural details are in Section .

4.2 Design of the GraphMet Training Objective

To learn a valid Koopman approximation without latent collapse, the objective must (i) align the encoder with leading Koopman singular modes, (ii) maintain an isotropic latent space so all modes remain active, and (iii) ensure accurate physical reconstruction via the decoder. While (i)–(ii) are enforced through spectral alignment, this loss is rotation-invariant, so (iii) requires an additional physical grounding term to resolve the ambiguity. This motivates the full objective design.

Covariance estimation. Given a batch of size N_B of one-step transitions $\{(\mathbf{x}^t_i, \mathbf{x}^{t+1}_i)\}_{i=1}^{N_B}$, we let

$$\hat{\mathbf{C}}_\phi(t) = \frac{1}{N_B} \sum_{i=1}^{N_B} \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_{t,i}) \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_{t,i})^\top \quad \text{and} \quad \hat{\mathbf{C}}_\phi(t, t+1) = \frac{1}{N_B} \sum_{i=1}^{N_B} \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_{t,i}) \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_{t+1,i})^\top. \tag{10}$$

These are $r \times r$ matrices regardless of the encoder width, keeping the objective computationally light.

Koopman predictability. The ratio

$$\mathcal{L}_{\text{spec}}(\boldsymbol{\phi}) \;=\; \frac{\|\hat{\mathbf{C}}_{\phi}(t, t+1)\|_{\text{HS}}^2}{\|\hat{\mathbf{C}}_{\phi}(t)\|_F \, \|\hat{\mathbf{C}}_{\phi}(t+1)\|_F} \tag{11}$$

measures how predictable the dynamics are in the learned latent basis. It is scale-invariant and lies in $[0, 1]$. This quantity is maximized when the encoder spans the leading singular subspace of $\mathcal{K}$ [14].

Isotropy penalty. To prevent the condition number of the latent covariance matrix from collapsing, we apply the following penalty:

$$R(\mathbf{C}) = \operatorname{tr}(\mathbf{C}^2 - \mathbf{C} - \ln \mathbf{C}). \tag{12}$$

Note that $R(\mathbf{C}) \geq 0$ with equality iff $\mathbf{C} = \mathbf{I}_r$. The penalty is applied to both $\hat{\mathbf{C}}_{\phi}(t)$ and $\hat{\mathbf{C}}_{\phi}(t+1)$.

Physical grounding. The spectral objective is invariant to orthogonal transformations of the latent space. Any rotation of the encoder's output leaves $\hat{\mathbf{C}}_{\phi}(t, t+1)$, $\hat{\mathbf{C}}_{\phi}(t)$, and $\hat{\mathbf{C}}_{\phi}(t+1)$ unchanged, but yields a different decoder. To break this variability, we anchor the decoder in physical space through a one-step forecast loss targeting the parameters $\boldsymbol{\phi}$ from (1) and the matrix $\mathbf{K}$ from (9),

$$\mathcal{L}_{\text{phys}}(\boldsymbol{\phi}, \mathbf{K}) = \frac{1}{N} \sum_i \left\| \boldsymbol{g}_{\boldsymbol{\theta}}\big(\mathbf{K} \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_i^t)\big) - \mathbf{x}_i^{t+1} \right\|_2^2. \tag{13}$$

Full objective. We jointly optimize the three components, weighted by $\xi_1, \xi_2 > 0$. The objective is

$$\mathcal{J}(\boldsymbol{\phi}, \mathbf{K}) = -\mathcal{L}_{\text{spec}}(\boldsymbol{\phi}) + \xi_1 \left[R(\hat{\mathbf{C}}_{\phi}(t)) + R(\hat{\mathbf{C}}_{\phi}(t+1)) \right] + \xi_2 \mathcal{L}_{\text{phys}}(\boldsymbol{\phi}, \mathbf{K}). \tag{14}$$

Theorem 4.1 (**Eckart–Young–Mirsky optimality, general compact case**). *Suppose the true Koopman operator $\mathcal{K}$ satisfies Assumption 2.1, with singular value decomposition $\mathcal{K} = \sum_{i \geq 1} \sigma_i \, u_i \otimes v_i$, $\sigma_1 \geq \sigma_2 \geq \cdots > 0$, so that $\mathcal{K} v_i = \sigma_i u_i$ and $\mathcal{K}^* u_i = \sigma_i v_i$. Assume $\sigma_r > \sigma_{r+1}$, where r is the latent mesh dimension. For any $\xi_1, \xi_2 > 0$, let $(\boldsymbol{\phi}^\star, \mathbf{K}^\star)$ denote the parameters minimizing $\mathcal{J}(\cdot, \cdot)$ in (14) subject to $\hat{\mathbf{C}}_{\boldsymbol{\phi}^\star}(t) = \hat{\mathbf{C}}_{\boldsymbol{\phi}^\star}(t+1) = \mathbf{I}_r$, and let $\mathcal{P}_{\boldsymbol{\phi}^\star}(t)$, $\mathcal{P}_{\boldsymbol{\phi}^\star}(t+1)$ be the orthogonal projectors onto the source and target observable subspaces spanned by the components of $\boldsymbol{f}_{\boldsymbol{\phi}^\star}(\cdot, t)$ and $\boldsymbol{f}_{\boldsymbol{\phi}^\star}(\cdot, t+1)$. The following assertions hold.*

- *(i)* $\mathcal{P}_{\boldsymbol{\phi}^\star}(t) \, \mathcal{K} \, \mathcal{P}_{\boldsymbol{\phi}^\star}(t+1) = \mathcal{K}_r := \sum_{i=1}^r \sigma_i u_i \otimes v_i$*; equivalently, the source subspace spans* $(u_i)_{i \leq r}$ *and the target subspace spans* $(v_i)_{i \leq r}$.
- *(ii) If* $\varsigma_1 \geq \cdots \geq \varsigma_r$ *denote the singular values of* $\mathbf{K}^\star$*, then* $\varsigma_i = \sigma_i$ *for all* $i \in \{1, \ldots, r\}$.
- *(iii)* $\|\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}^\star}(t) \mathcal{K} \mathcal{P}_{\boldsymbol{\phi}^\star}(t+1)\|_{\text{HS}}^2 = \sum_{i=r+1}^{\infty} \sigma_i^2$*, which is the Eckart–Young–Mirsky lower bound over all rank-r approximations of $\mathcal{K}$.*

The statement holds for every compact $\mathcal{K}$: it is obtained from the singular value decomposition and finite-rank approximation of Hilbert–Schmidt operators, and it keeps the source and target observable subspaces distinct, exactly as the algorithm does through $\hat{\mathbf{C}}_{\boldsymbol{\phi}}(t)$ and $\hat{\mathbf{C}}_{\boldsymbol{\phi}}(t+1)$ in (11). Normality, self-adjointness and reversibility of the dynamics are therefore not required anywhere in the derivation.

The proof is in Section A.6. It reveals that the spectral terms in the objective (14) determine which subspace is learned, while $\mathcal{L}_{\text{phys}}$ determines how that subspace maps back to physical space. Successfully optimizing the GraphMet objective guarantees both a spectrally optimal Koopman approximation and a physically grounded model.

Extended Probabilistic Forecasting Since Theorem 4.1 is stated without any normality assumption, the left and right singular structures of $\mathcal{K}$ must be treated separately throughout, which is also what makes probabilistic forecasting available in closed form. The approximate Koopman matrix has an SVD of the form $\mathbf{K} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$, with $\mathbf{S} = \operatorname{diag}(\varsigma_1, \ldots, \varsigma_r)$. The corresponding left and right singular functions, evaluated at an embedded state $\mathbf{z} = \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x})$, are $u_i(\mathbf{z}) = [\mathbf{U}^\top \mathbf{z}]_i$ and $v_i(\mathbf{z}) = [V^\top \mathbf{z}]_i$. By explicitly recovering these functions and their temporal decay rates $(\varsigma_i)_{1 \leq i \leq r}$, GraphMet natively

integrates with the Neural Conditional Probability (NCP) framework [15]. For any target event $B \in \Sigma$, the transition probability at lead time T can be approximated using the rank r truncation:

$$\mathbb{P}[\mathbf{z}^{t+T} \in B \mid \mathbf{z}^t = \mathbf{z}] \approx \int_B \left(1 + \sum_{i=1}^{r} \varsigma_i^T u_i(\mathbf{z}) v_i(\mathbf{z}')\right) \mu(\mathrm{d}\mathbf{z}').$$

By integrating appropriate test functions against the underlying density, quantities such as conditional variances and probabilistic tracks can be computed easily, bypassing the need for computationally expensive ensemble generation.

5 Related Work

Oversmoothing in Global-to-Regional Weather Forecasting. GraphCast [17] introduced an encode–process–decode architecture on an icosahedral mesh using a deep GNN processor, while AIFS [18] replaces the processor with a transformer. Both are trained with area-weighted MSE and show reduced forecast activity with lead time, consistent with progressive loss of fine-scale structure. This behaviour is linked to oversmoothing in deep message-passing networks [21, 24], where repeated propagation homogenizes latent representations. We identify this as a general limitation of deep weather processors, largely independent of whether the backbone is a GNN or transformer. Recent Koopman-based approaches instead model dynamics through linear latent operators, improving stability and scalability by avoiding deep propagation. Oversmoothing is especially harmful in regional forecasting due to boundary-condition sensitivity [25, 1], motivating controlled design.

Koopman operators. The Koopman operator formalism originates from the works of Koopman and von Neumann in the 1930s [11, 12]. It lifts nonlinear dynamics into a function space where evolution becomes linear. Practical approximations include dynamic mode decomposition (DMD) [32, 35, 38], which extract spectral structure directly from data. Early approaches relied on hand-crafted observables [3], enabling applications such as robotic control, but often with limited generalization. More recent methods learn Koopman-invariant embeddings using neural networks [23, 36], yielding more expressive latent spaces and improved performance in complex systems such as fluid dynamics. The VAMP framework [15, 14] provides a variational formulation of Koopman approximation error that underlies modern training objectives. Our approach differs by explicitly modeling compositional structure with graph-based operators, improving generalization to systems with variable numbers of interacting components or heterogeneous geometries.

Probabilistic Weather Forecasting. GenCast [28], GraphEFM [26], and DiffLAM [20] extend deterministic forecasting models with diffusion-based approaches to generate ensemble forecasts. However, they inherit the spectral smoothing of the underlying processor, which remains unchanged. Similarly, AIFS-CRPS [19] retains the deterministic AIFS encoder–processor–decoder architecture, but replaces deterministic training with a probabilistic objective based on the Continuous Ranked Probability Score (CRPS), enabling direct prediction of forecast distributions rather than single trajectories. This training is computationally expensive, as CRPS estimation requires sampling multiple ensemble members during optimisation.

6 Experiments

6.1 Setup and Implementation Details

Data. We use ERA5 [9] (0.25°, 6-hourly, 2000–2019 train, 2022 test) for global pretraining, and CERRA [31] (5.5 km, 30°N–75°N, 35°W–45°E, 2010–2020 train, 2022 test) for regional fine-tuning. A preliminary stage on ERA5 dates in 1940–2000 selects (ξ_1, ξ_2) only, 2020–2021 is held out for validation and Koopman re-estimation, and the case studies are inference-only; no evaluation period overlaps a training window (Table 14).

Baselines and Metrics. Global baselines include GraphCast, Pangu-Weather, and ForeCastNet [27]. Regional baselines are GraphFM, GraphEFM [26], and DiffLAM [20]. We report latitude-weighted RMSE and ACC [30], CRPS [8], Activity (forecast-to-observed variance ratio, ideal 1) and Spectral Energy Ratio (SER, ideal 1 over mid-wavenumbers).

Training optimization. We use AdamW with cosine decay from 10^{-4} to 10^{-6} over 500k steps, effective batch size 64 and $\gamma = 10^{-3}$ on 64 A100 GPUs; Stage 2 fine-tunes the regional decoder for

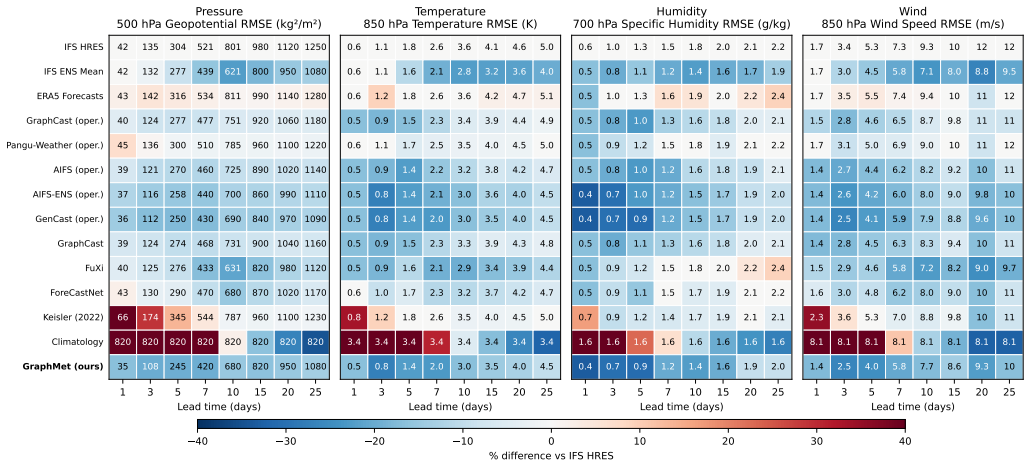


Figure 2: Relative RMSE (%) versus IFS HRES [18] over 1–25 forecast days for 500 hPa geopotential, 850 hPa temperature, 700 hPa humidity, and 850 hPa wind speed. Blue indicates lower error and red higher error. GraphMet achieves competitive or improved skill across most variables and lead times compared to physical and ML baselines.

50k steps at 5×10^{-5} with a frozen global encoder. Computational costs are reported in Section E.1. The 14,336 parameters quoted for GraphMet are the Koopman processor alone; the full model holds 20.7 M against 36.0 M (GraphCast) and 57.5 M (GenCast), with matched FLOPs and peak memory in Table 7.

6.2 Ensemble Forecasting with Neural Conditional Probability Scores

CRPS is evaluated in closed form from the analytic NCP density above, not by 50 independent rollouts; the 50-member fields in Figure 1C are draws from that single learned distribution, scored with the baselines' own empirical estimator for a budget-matched comparison. On CERRA $t2m$ at day 5 the closed-form and 50-sample values agree within Monte Carlo error (1.14 vs. 1.16, against 1.33 for DiffLAM), at $\sim 228\times$ lower cost (Section D.4).

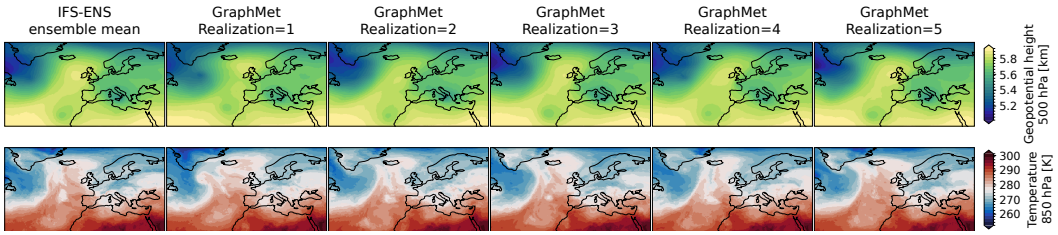


Figure 3: Ensemble realizations from GraphMet-ENS (cycle 2025-01-01 06 UTC, valid +102 h): three of five members drawn from the single learned conditional density, against the 51-member IFS-ENS mean, for z500 (top) and t850 (bottom). The members keep gradients the mean averages away; all five members globally in Figure 9.

6.3 Global to Regional Weather Modeling

GraphMet improves both RMSE and forecast sharpness across variables and lead times, as shown in Figure 2. In most ML weather prediction systems higher activity costs RMSE, since phase errors are penalized quadratically (Figure 1-B). In the regional setting, GraphMet avoids boundary artifacts by using a shared global encoder–decoder that directly produces CERRA-resolution outputs (Figure 12). GraphEFM instead oversmooths fine-scale structure, consistent with the 11% and 14% RMSE gains at 5 km and the better SSR in Figure 5.

6.4 Ablation Study

We ablate each design decision in GraphMet, reporting latitude-weighted RMSE at 72 h and 120 h for z500 ($m^2 s^{-2}$) and u850 (ms^{-1}), together with $\lambda^+_{\min}$ defined in Section 3, which quantifies latent isotropy. All variants share the same encoder, decoder, mesh, and data.

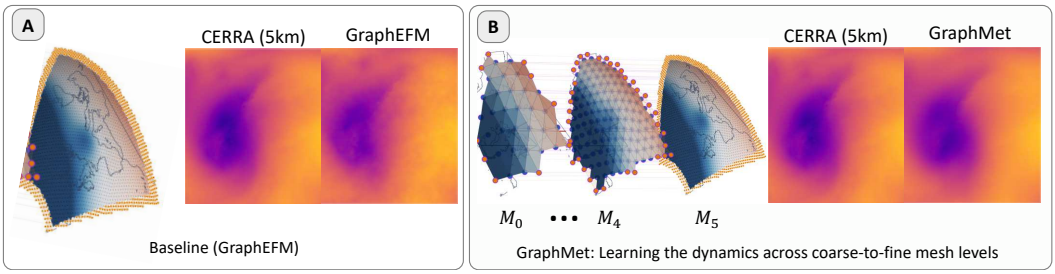


Figure 4: Global-to-regional module: the Koopman operator acts across the icosahedral levels M_0–M_6, so no lateral boundary condition is prescribed.

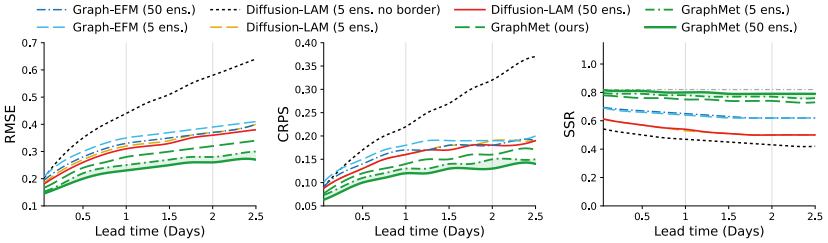


Figure 5: Forecast skill over 0–2.5 days across global and regional settings. **Left/centre:** RMSE and CRPS (lower is better). **Right:** SSR, with grey marking perfect calibration and the green band the gain from 5 to 50 members. Diffusion-LAM without boundary conditions degrades after 1 day, which the multi-scale Koopman operator avoids.

A1 - What is the impact of the Koopman processor? As shown in Table 1, increasing the depth of the message-passing processor strongly reduces $\lambda^{+}_{\min}$, independently of the training objective, indicating that spectral losses alone cannot prevent structural averaging. In contrast, a single Koopman step consistently outperforms all multi-step variants. This is expected: repeated applications of K contract spatial variance by a factor related to $\lambda_{\max}(K)^2$, reintroducing the smoothing the architecture aims to remove. Diagnostics: Sections D.2 and D.3.

A2 - How do different components of the model affect performance? Three findings emerge from Table 17. First, MSE alone collapses $\lambda^{+}_{\min}$ to 1.1×10^{-2}, close to the deep processor, indicating strong latent degeneracy under purely reconstruction-based training. Second, removing $\mathcal{L}_{phys}$ (Spectral+Isotropy) improves isotropy but degrades forecast skill, showing that spectral alignment without physical grounding leaves the decoder under-constrained. Finally, replacing climatological weighting with a uniform ℓ_2 loss recovers most but not all performance, with a remaining gap of $\Delta\text{z500}-120\text{h} = 28, \text{m}^2\text{s}^{-2}$, suggesting that the weighting structure not just the loss itself is important to prevent mesoscale suppression.

7 Conclusion and Limitations

We investigate the limitations of deep message-passing architectures in weather prediction and introduce a Koopman-based formulation that preserves latent spectral structure over long horizons. While repeated message passing smooths latent representations and suppresses fine-scale dynamics, GraphMet evolves learned spectral modes linearly, reducing computational cost while preserving spatial detail. This enables accurate regional forecasting at $5\,\text{km}$ resolution with lower inference cost, highlighting spectral operator learning as an efficient alternative to deep message passing.

Limitations. GraphMet uses a fixed multiscale icosahedral mesh, requiring architectural adaptations for dynamic regional discretizations. Evaluation is limited to European CERRA data, with (ξ_1, ξ_2) selected by validation following preliminary ERA5 experiments from 1940–2000 (Sections C and D.6). Moreover, Theorem 3.2 may yield loose bounds with residual connections; the analytical density approximation degrades when the discarded spectral tail is significant; parameter count remains a consideration (Table 7); and two extreme-weather case studies cannot establish generalizable extreme-event forecasting skill.

Acknowledgments and Disclosures

We thank the anonymous reviewers and the Area Chair of NeurIPS 2026 for a careful and constructive review. This work was granted access to the HPC resources of IDRIS under the allocation AD011014921 made by GENCI (Grand Equipement National de Calcul Intensif), and was supported by OneTech, TotalEnergies, which provided the computational infrastructure. We thank the European Centre for Medium-Range Weather Forecasts (ECMWF) and the Copernicus Climate Change Service (C3S) for open access to the ERA5 and CERRA reanalyses, and the ECMWF for public access to the operational IFS and IFS-ENS forecasts used as numerical references. We thank the WeatherBench 2 team for maintaining the evaluation protocol and baseline scores that make comparison across this literature possible. Part of this work was carried out while Khalid Oublal was supported by a doctoral fellowship at Institut Polytechnique de Paris, and by an Individual Fellowship from Hi! Paris, which supported an earlier stage of the project. Vladimir R Kostic received funding from NextGenerationEU and Provincial Secretariat for Higher Education and Scientific Research of Vojvodina (Research Grant 003870560 2025 09418 003 000 000 001 04 004). Karim Lounici received funding from the European Union's Horizon Europe research and innovation program under grant agreement 101120237 (ELIAS) and Fondation de l'Ecole Polytechnique Hi! PARIS.

References

[1] S. Adamov et al. Building machine learning limited area models. *arXiv:2504.09340*, 2025.

[2] 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.

[3] Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLOS ONE*, 11(2):e0150171, 2016.

[4] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

[5] Alexandre J. Chorin, Ole H. Hald, and Raz Kupferman. Optimal prediction and the mori–zwanzig representation of irreversible processes. *Proceedings of the National Academy of Sciences*, 97(7):2968–2973, 2000.

[6] John B Conway. *A course in functional analysis*. Springer, 2019.

[7] Dror Givon, Raz Kupferman, and Andrew Stuart. Extracting macroscopic dynamics: model problems and algorithms. *Nonlinearity*, 17(6):R55–R127, 2004.

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

[9] Hans Hersbach et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[10] Mason Kamb, Eurika Kaiser, Steven L. Brunton, and J. Nathan Kutz. Time-delay observables for koopman: theory and applications. *SIAM Journal on Applied Dynamical Systems*, 19(2):886–917, 2020.

[11] Bernard O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[12] Bernard O. Koopman and John von Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, 1932.

[13] Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018.

[14] Vladimir Kostic, Karim Lounici, Pietro Novelli, and Massimiliano Pontil. Sharp spectral rates for koopman operator learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 32328–32339. Curran Associates, Inc., 2023.

[15] Vladimir Kostic, Pietro Novelli, Andreas Maurer, Carlo Ciliberto, Lorenzo Rosasco, and Massimiliano Pontil. Learning dynamical systems via koopman operator regression in reproducing kernel hilbert spaces. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 4017–4031. Curran Associates, Inc., 2022.

[16] Vladimir R. Kostic, Karim Lounici, Pietro Novelli, and Massimiliano Pontil. Sharp spectral rates for koopman operator learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[17] 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.

[18] 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*, 2024.

[19] 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. *npj Artificial Intelligence*, 2(1):18, 2026.

[20] Erik Larsson et al. DiffLAM: Probabilistic limited area weather forecasting with diffusion. *arXiv:2502.07532*, 2025.

[21] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[22] Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional koopman operators for model-based control. *arXiv preprint arXiv:1910.08264*, 2019.

[23] Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.

[24] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

[25] Joel Oskarsson et al. Graph-based neural weather prediction for limited area modelling. *ICML AI4Science Workshop*, 2023.

[26] Joel Oskarsson et al. Probabilistic weather forecasting with hierarchical graph neural networks. In *NeurIPS*, 2024.

[27] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[28] 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.

[29] Florence Rabier, Heikki Järvinen, Erik Klinker, Jean-François Mahfouf, and Adrian Simmons. The ECMWF operational implementation of four-dimensional variational assimilation. Part I: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1143–1170, 2000.

[30] Stephan Rasp et al. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16:e2023MS004019, 2024.

[31] Martin Ridal et al. CERRA, the Copernicus European Regional ReAnalysis system. *Quarterly Journal of the Royal Meteorological Society*, 2024.

[32] Clarence W. Rowley et al. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.

[33] Tim Sauer, James A. Yorke, and Martin Casdagli. Embedology. *Journal of Statistical Physics*, 65(3–4):579–616, 1991.

[34] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[35] Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[36] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *NeurIPS*, 2017.

[37] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381. Springer, 1981.

[38] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[39] Robert Zwanzig. *Nonequilibrium Statistical Mechanics*. Oxford University Press, 2001.

Supplementary Material

This appendix is organised as follows. We first present the theoretical proofs, followed by additional implementation details, hyperparameter sensitivity analyses, and extended experimental results. We then provide case studies on extreme weather events and cyclone tracking, before concluding with further qualitative visualisations.

Contents

A Proofs **15**

A.1 Proof of Theorem 3.2 . . . 15
A.2 Proof of Corollary 3.3 . . . 15
A.3 Proof of Proposition 3.1 . . . 15
A.4 Covariance Geometry and the Role of Latent Conditioning . . . 16
A.5 Koopman operator regression versus Mori–Zwanzig reduction . . . 16
A.6 Proof of Theorem 4.1 . . . 17

B Architecture and Training **17**

B.1 ERA5 Variable Specification . . . 18

C Hyperparameter Sensitivity **18**

C.1 Extreme Weather Case Study with GraphMet: Tropical Cyclone Narelle . . . 18

D Cost Accounting and Additional Diagnostics **20**

D.1 Full Model Cost: Parameters, FLOPs and Memory . . . 20
D.2 Dirichlet-Energy Contraction in Deployed Processors . . . 20
D.3 Lead-Time-Resolved and Region-Conditioned Mesoscale Diagnostics . . . 21
D.4 Probabilistic Mechanism: Analytic NCP versus Sampled Ensembles . . . 21
D.5 Datasets, Licensing and Preprocessing . . . 22
D.6 Complete Data Provenance and Splits . . . 22
D.7 Independent Case Study: Storm Goretti, 8–9 January 2026 . . . 24
D.8 Ensemble Realizations and Regional Maps . . . 25
D.9 Loss-Term Ablation Table . . . 25
D.10 Zonal Power Spectra . . . 25

E Extended Results **26**

E.1 Inference Cost . . . 26
E.2 From Global to Regional Modeling with CERRA . . . 27

F GraphMet Global Forecast Visualizations **30**

A Proofs

A.1 Proof of Theorem 3.2

Proof. Define the projector $\mathbf{P} = N_M^{-1}\mathbf{1}\mathbf{1}^\top$, where $\mathbf{1} = (1, \dots, 1)^\top$, and its orthogonal $\mathbf{P}^\perp = \mathbf{I} - \mathbf{P}$. At any layer ℓ, let $\overline{\mathbf{H}}^{(\ell)} = \mathbf{P}\mathbf{H}^{(\ell)}$ and $\overline{\mathbf{H}}_\perp^{(\ell)} = \mathbf{P}^\perp\mathbf{H}^{(\ell)}$, so that $\mathbf{H}^{(\ell)} = \overline{\mathbf{H}}^{(\ell)} + \overline{\mathbf{H}}_\perp^{(\ell)}$. Since $\mathbf{A}$ is row-stochastic, we have $\mathbf{A}\mathbf{1} = \mathbf{1}$. Therefore, $\mathbf{A}\overline{\mathbf{H}}^{(\ell)} = \mathbf{A}(N_M^{-1}\mathbf{1}\mathbf{1}^\top\mathbf{H}^{(\ell)}) = N_M^{-1}\mathbf{1}\mathbf{1}^\top\mathbf{H}^{(\ell)} = \overline{\mathbf{H}}^{(\ell)}$. Using the Perron-Frobenius theorem and the fact that $\mathbf{A}$ is a symmetric matrix, we deduce that $\mathbf{A}$ leaves the space $V_\perp = \{v \in \mathbb{R}^{N_M} \,:\, v^\top\mathbf{1} = 0\}$ invariant and $\|\mathbf{A}v\|_2 \leq \lambda_2\|v\|_2$ for all $v \in V_\perp$. In particular, considering the matrix $\overline{\mathbf{H}}_\perp^{(\ell)}$ column-by-column, we have

$$\|\mathbf{A}\overline{\mathbf{H}}_\perp^{(\ell)}\|_F \leq \lambda_2\|\overline{\mathbf{H}}_\perp^{(\ell)}\|_F. \tag{15}$$

Using the expression $\|\overline{\mathbf{H}}_\perp^{(\ell+1)}\|_F = \min_{\mathbf{H}\in\mathcal{Z}}\|\mathbf{H}^{(\ell+1)} - \mathbf{P}\mathbf{H}\|_F$, we deduce the upper-bound

$$\|\overline{\mathbf{H}}_\perp^{(\ell+1)}\|_F \leq \|\gamma(\mathbf{A}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}) - \gamma(\mathbf{A}\overline{\mathbf{H}}^{(\ell)}\mathbf{W}^{(\ell)})\|_F.$$

Because γ is applied element-wise and is s-Lipschitz continuous, we get

$$\|\overline{\mathbf{H}}_\perp^{(\ell+1)}\|_F \leq s\|\mathbf{A}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)} - \mathbf{A}\overline{\mathbf{H}}^{(\ell)}\mathbf{W}^{(\ell)}\|_F = s\|\mathbf{A}\overline{\mathbf{H}}_\perp^{(\ell)}\mathbf{W}^{(\ell)}\|_F.$$

By sub-multiplicativity of the Frobenius norm, the assumption $\|\mathbf{W}^{(\ell)}\|_{\mathrm{op}} \leq 1$ and (15) imply

$$\|\overline{\mathbf{H}}_\perp^{(\ell+1)}\|_F \leq s\|\mathbf{A}\overline{\mathbf{H}}_\perp^{(\ell)}\|_F\|\mathbf{W}^{(\ell)}\|_{\mathrm{op}} \leq s\lambda_2\|\overline{\mathbf{H}}_\perp^{(\ell)}\|_F.$$

Iterating this inequality across all L layers yields

$$\|\overline{\mathbf{H}}_\perp^{(L)}\|_F \leq (s\lambda_2)^L\|\overline{\mathbf{H}}_\perp^{(0)}\|_F. \tag{16}$$

Now, remark that $E(\mathbf{H}^{(L)}) = \mathrm{tr}((\overline{\mathbf{H}}_\perp^{(L)})^\top(\mathbf{I} - \mathbf{A})\overline{\mathbf{H}}_\perp^{(L)})$. By applying the Courant-Fischer theorem to the Laplacian $\mathbf{I} - \mathbf{A}$, we get $E(\mathbf{H}^{(L)}) \leq \lambda_{\max}(\mathbf{I} - \mathbf{A})\|\overline{\mathbf{H}}_\perp^{(L)}\|_F^2$, where $\lambda_{\max}(\mathbf{I} - \mathbf{A})$ is the largest eigenvalue of the Laplacian. Substituting (16) gives

$$E(\mathbf{H}^{(L)}) \leq \lambda_{\max}(\mathbf{I} - \mathbf{A})(s\lambda_2)^{2L}\|\overline{\mathbf{H}}_\perp^{(0)}\|_F^2. \tag{17}$$

Using the Courant-Fischer theorem again in the subspace $V_\perp$, letting $\lambda_{\min}^+(\mathbf{I} - \mathbf{A}) = 1 - \lambda_2$ denote the smallest positive eigenvalue of the Laplacian, we have $E(\mathbf{H}^{(0)}) \geq \lambda_{\min}^+(\mathbf{I} - \mathbf{A})\|\overline{\mathbf{H}}_\perp^{(0)}\|_F^2$, and therefore $\|\overline{\mathbf{H}}_\perp^{(0)}\|_F^2 \leq E(\mathbf{H}^{(0)})/\lambda_{\min}^+(\mathbf{I} - \mathbf{A})$. Combining this with (17) concludes the proof. □

A.2 Proof of Corollary 3.3

Proof. For any L, we can find $c_L \in \mathbb{R}^m$ such that $\mathbf{H}^{(L)} = \mathbf{1}c_L^\top + \overline{\mathbf{H}}_\perp^{(L)}$. The latent covariance matrix expands as $\mathbf{C}_{\boldsymbol{\phi}}^{(L)} = \mathbb{E}[\|c_L\|_2^2]\mathbf{1}\mathbf{1}^\top + \mathbf{R}_L$, where the residual $\mathbf{R}_L$ is given by

$$\mathbf{R}_L = \mathbb{E}[\mathbf{1}c_L^\top(\overline{\mathbf{H}}_\perp^{(L)})^\top + \overline{\mathbf{H}}_\perp^{(L)}c_L\mathbf{1}^\top + \overline{\mathbf{H}}_\perp^{(L)}(\overline{\mathbf{H}}_\perp^{(L)})^\top].$$

From the proof of Theorem 3.2, we have $\|\overline{\mathbf{H}}_\perp^{(L)}\|_F \leq (s\lambda_2)^L\|\overline{\mathbf{H}}_\perp^{(0)}\|_F$. Therefore, using the triangle inequality, we obtain $\|\mathbf{R}_L\|_F = \mathcal{O}((s\lambda_2)^L)$. As L goes to infinity, only the rank-1 component remains in the latent covariance matrix $\mathbf{C}_\phi^{(L)}$. Therefore, $\lambda_{\min}^+(\mathbf{C}_{\boldsymbol{\phi}}^{(L)})$ converges to zero as L goes to infinity. □

A.3 Proof of Proposition 3.1

Proof. Let $\mathcal{P}_{\boldsymbol{\phi}}$ be the $L^2(\mathcal{Z}, \mu)$-orthogonal projector onto $V_{\boldsymbol{\phi}} = \mathrm{span}(\boldsymbol{f}_{\boldsymbol{\phi}})$ and $\mathcal{P}_{\boldsymbol{\phi}}^\perp = \mathcal{I} - \mathcal{P}_{\boldsymbol{\phi}}$. Since $\mathcal{P}_{\boldsymbol{\phi}} + \mathcal{P}_{\boldsymbol{\phi}}^\perp = \mathcal{I}$,

$$\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}} = (\mathcal{I} - \mathcal{P}_{\boldsymbol{\phi}})\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}} + \mathcal{K}(\mathcal{I} - \mathcal{P}_{\boldsymbol{\phi}}) = \mathcal{P}_{\boldsymbol{\phi}}^\perp\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}} + \mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}^\perp. \tag{18}$$

The two terms are orthogonal in the Hilbert–Schmidt inner product. Indeed, using $\langle A, B\rangle_{\mathrm{HS}} = \mathrm{tr}(A^*B)$, the idempotence and self-adjointness of the projectors, and $\mathcal{P}_{\boldsymbol{\phi}}\mathcal{P}_{\boldsymbol{\phi}}^{\perp} = 0$,

$$\langle \mathcal{P}_{\boldsymbol{\phi}}^{\perp}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}, \mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\rangle_{\mathrm{HS}} = \mathrm{tr}\big(\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}^*\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\big) = \mathrm{tr}\big(\mathcal{K}^*\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\mathcal{K}\,\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\mathcal{P}_{\boldsymbol{\phi}}\big) = 0,$$

where the last equality uses the trace cyclicity and $\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\mathcal{P}_{\boldsymbol{\phi}} = 0$. Applying the Pythagorean identity to (18) gives the first equality of (6),

$$\|\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2 = \|\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2 + \|\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\|_{\mathrm{HS}}^2.$$

For the second equality, decompose $\mathcal{K} = \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}} + (\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}})$ and note that $\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}$ is Hilbert–Schmidt orthogonal to each term of (18), since $\mathrm{tr}(\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}^*\mathcal{P}_{\boldsymbol{\phi}}\mathcal{P}_{\boldsymbol{\phi}}^{\perp}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}) = 0$ and $\mathrm{tr}(\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}^*\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}^{\perp}) = 0$. Hence $\|\mathcal{K}\|_{\mathrm{HS}}^2 = \|\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2 + \|\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2$, which rearranges to $\|\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2 = \|\mathcal{K}\|_{\mathrm{HS}}^2 - \|\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2$. Under Assumption 2.1, $\|\mathcal{K}\|_{\mathrm{HS}} < \infty$ and does not depend on $\boldsymbol{\phi}$, so minimizing the left-hand side over $\boldsymbol{\phi}$ is equivalent to maximizing $\|\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2$. The argument uses no normality hypothesis, no auxiliary norm and no spectral constant. □

A.4 Covariance Geometry and the Role of Latent Conditioning

We expand on Section 3.1. With Φ_τ the dictionary of observables spanned by the components of $\boldsymbol{f}_{\boldsymbol{\phi}}(\cdot, \tau)$ and c a coefficient vector, the latent covariance of (5) is exactly the Gram matrix of that dictionary,

$$\|\Phi_\tau c\|_{L^2(\mu_\tau)}^2 = c^\top \mathbf{C}_{\boldsymbol{\phi}}(\tau)\, c, \qquad \mathcal{P}_\tau = \Phi_\tau\, \mathbf{C}_{\boldsymbol{\phi}}(\tau)^\dagger\, \Phi_\tau^*, \tag{19}$$

Whitening both dictionaries defines the normalized operator

$$\overline{\mathbf{K}}_{\tau,\Delta} := \mathbf{C}_{\boldsymbol{\phi}}(\tau)^{-1/2}\, \mathbf{C}_{\boldsymbol{\phi}}(\tau, \tau+\Delta)\, \mathbf{C}_{\boldsymbol{\phi}}(\tau+\Delta)^{-1/2}, \qquad \|\mathcal{P}_\tau\mathcal{K}_\tau\mathcal{P}_{\tau+\Delta}\|_{\mathrm{HS}}^2 = \|\overline{\mathbf{K}}_{\tau,\Delta}\|_F^2, \tag{20}$$

with $(\cdot)^\dagger$ the Moore–Penrose pseudo-inverse. The identity $\|\mathcal{P}_\tau\mathcal{K}_\tau\mathcal{P}_{\tau+\Delta}\|_{\mathrm{HS}}^2 = \|\overline{\mathbf{K}}_{\tau,\Delta}\|_F^2$ of (20) follows by expressing $\mathcal{P}_\tau\mathcal{K}_\tau\mathcal{P}_{\tau+\Delta}$ in the orthonormal bases obtained by whitening each dictionary.

Writing $\mathbf{C}_{\boldsymbol{\phi}} := \mathbf{C}_{\boldsymbol{\phi}}(t) = \mathbf{C}_{\boldsymbol{\phi}}(t+1)$ by stationarity and $\overline{\mathbf{K}} = \mathbf{C}_{\boldsymbol{\phi}}^{-1/2}\mathbf{C}_{\boldsymbol{\phi}}(t, t{+}1)\mathbf{C}_{\boldsymbol{\phi}}^{-1/2}$, submultiplicativity gives $\|\overline{\mathbf{K}}\|_F \le \|\mathbf{C}_{\boldsymbol{\phi}}^{-1/2}\|_{\mathrm{op}}^2\|\mathbf{C}_{\boldsymbol{\phi}}(t, t{+}1)\|_F = \|\mathbf{C}_{\boldsymbol{\phi}}(t, t{+}1)\|_F/\lambda_{\min}^+(\mathbf{C}_{\boldsymbol{\phi}})$, and conversely $\|\mathbf{C}_{\boldsymbol{\phi}}(t, t{+}1)\|_F = \|\mathbf{C}_{\boldsymbol{\phi}}^{1/2}\overline{\mathbf{K}}\mathbf{C}_{\boldsymbol{\phi}}^{1/2}\|_F \le \lambda_{\max}(\mathbf{C}_{\boldsymbol{\phi}})\|\overline{\mathbf{K}}\|_F$. Squaring both,

$$\frac{\|\mathbf{C}_{\boldsymbol{\phi}}(t, t{+}1)\|_F^2}{\lambda_{\max}(\mathbf{C}_{\boldsymbol{\phi}})^2} \;\le\; \|\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2 \;\le\; \frac{\|\mathbf{C}_{\boldsymbol{\phi}}(t, t{+}1)\|_F^2}{\lambda_{\min}^+(\mathbf{C}_{\boldsymbol{\phi}})^2}, \tag{21}$$

so the measurable cross-covariance pins down the captured energy of Proposition 3.1 only up to the factor $\kappa(\mathbf{C}_{\boldsymbol{\phi}})^2$. Independently of this bound, $\lambda_{\min}^+(\mathbf{C}_{\boldsymbol{\phi}}) \to 0$ means the latent representation becomes effectively rank-deficient: $\mathcal{P}_{\boldsymbol{\phi}}$ then has effective rank $r_{\mathrm{eff}} < r$ and $\|\mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}\|_{\mathrm{HS}}^2 \le \sum_{i \le r_{\mathrm{eff}}} \sigma_i^2$ by Theorem 4.1(iii). The directions lost first are those carrying the smallest σ_i, which are precisely the short-lived mesoscale modes that regional forecasting depends on. This is the mechanism that Corollary 3.3 shows deep message passing to trigger.

Finally, note that the invariance of (20) under $\boldsymbol{f}_{\boldsymbol{\phi}} \mapsto \lambda \boldsymbol{f}_{\boldsymbol{\phi}}$ is what makes the objective (11) well posed without any normalization constraint on $\boldsymbol{f}_{\boldsymbol{\phi}}$: the cross-covariance scales as λ^2 and the two whitening factors as λ^{-1} each.

A.5 Koopman operator regression versus Mori–Zwanzig reduction

The term $\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}^{\perp}$ in Proposition 3.1 plays a role reminiscent of the memory term that appears in a Mori–Zwanzig (MZ) analysis. The analogy is apt in that both objects describe the effect of unresolved directions, but the two constructions are mathematically different and the operational conclusions differ, so we separate them here.

MZ reduction [39, 5, 7] starts from a known evolution equation, selects a set of resolved variables, and derives a self-contained equation in those variables. Because the resolved set is not dynamically closed, the exact reduced equation is non-Markovian and decomposes into a Markovian term, a Volterra memory kernel and a fluctuation term. Memory appears because closure in the resolved variables is imposed.

Transfer-operator regression [13, 15, 16] does not close an equation. It approximates the linear operator $\mathcal{K}$ on a function space by a finite-rank operator and measures the residual in an operator norm. Here $\mathcal{P}_{\boldsymbol{\phi}}$ is an $L^2(\mu_\tau)$-orthogonal projection onto a subspace of *observables*, not a projection of physical state variables, and $\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}}\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}$ is the finite-rank approximation residual. Since no reduced state equation is derived and no closure is performed, no memory kernel is generated, and the identity (6) is exact with no additional terms.

Memory is instead introduced in GraphMet by design, through state augmentation. We predict $\mathbf{x}^{t+\Delta}$ from $\tilde{\mathbf{x}}^t = (\mathbf{x}^{t-\Delta}, \mathbf{x}^t)$, a 12-hour context with $\Delta = 6$ h, together with dynamic periodic forcings (day of year, hour of day, top-of-atmosphere solar radiation) and static forcings (surface geopotential, land–sea mask). On this augmented space the dynamics are Markov, so a memoryless operator is the appropriate object; delay coordinates recover the state when observations are incomplete [37, 33, 10]. Our ablations show that a one-state context underperforms the two-state context, while additional history gives limited further gain, consistent with what is reported for GraphCast.

A.6 Proof of Theorem 4.1

Proof. We treat each point individually.

(i) Under the constraint $\hat{\mathbf{C}}_{\boldsymbol{\phi}^\star}(t) = \hat{\mathbf{C}}_{\boldsymbol{\phi}^\star}(t+1) = \mathbf{I}_r$, the denominator of $\mathcal{L}_{\text{spec}}$ in (11) equals $\|\mathbf{I}_r\|_F^2 = r$, a constant, so minimizing $-\mathcal{L}_{\text{spec}}$ is equivalent to maximizing $\|\hat{\mathbf{C}}_{\boldsymbol{\phi}}(t, t+1)\|_{\text{HS}}^2$. Under the same whitening constraint, (19) and (20) give $\|\hat{\mathbf{C}}_{\boldsymbol{\phi}}(t, t+1)\|_{\text{HS}}^2 = \|\mathcal{P}_{\boldsymbol{\phi}}(t)\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}}(t+1)\|_{\text{HS}}^2$. By the variational characterisation of the singular value decomposition of a Hilbert–Schmidt operator (the VAMP-2 principle, 15), for any pair of rank-r orthogonal projectors $\mathcal{P}_{,,}$

$$\|\mathcal{P}\mathcal{K}\|_{\text{HS}}^2 \;\le\; \sum_{i=1}^{r} \sigma_i^2,$$

with equality if and only if $\text{ran}(\mathcal{P}) = \text{span}(u_1, \dots, u_r)$ and $\text{ran}() = \text{span}(v_1, \dots, v_r)$. The strict spectral gap $\sigma_r > \sigma_{r+1}$ makes these two maximising subspaces unique. At the optimum $\boldsymbol{\phi}^\star$ the bound is attained, hence $\mathcal{P}_{\boldsymbol{\phi}^\star}(t)$ projects onto the leading r left singular functions and $\mathcal{P}_{\boldsymbol{\phi}^\star}(t+1)$ onto the leading r right singular functions, and therefore $\mathcal{P}_{\boldsymbol{\phi}^\star}(t)\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}^\star}(t+1) = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i = \mathcal{K}_r$. Note that no relation between the u_i and the v_i is used; if $\mathcal{K}$ happens to be normal the two subspaces coincide up to phase.

(ii) Since $\mathbf{K}^\star$ represents the restriction of $\mathcal{K}$ to the pair of subspaces above, expressed in the orthonormal bases produced by the whitening constraint, its singular values coincide with the leading singular values of $\mathcal{K}$, giving (ii).

(iii) By Proposition 3.1 and part (i), $\|\mathcal{K} - \mathcal{P}_{\boldsymbol{\phi}^\star}(t)\mathcal{K}\mathcal{P}_{\boldsymbol{\phi}^\star}(t+1)\|_{\text{HS}}^2 = \|\mathcal{K}\|_{\text{HS}}^2 - \|\mathcal{K}_r\|_{\text{HS}}^2 = \sum_{i\ge 1} \sigma_i^2 - \sum_{i=1}^{r} \sigma_i^2 = \sum_{i=r+1}^{\infty} \sigma_i^2$, which is the Eckart–Young–Mirsky lower bound over all rank-r approximations of $\mathcal{K}$ and is therefore optimal.

□

B Architecture and Training

The core of the architecture follows the multi-mesh construction of [17, Section 3.3], based on a regular icosahedron refined six times to obtain 40 962 mesh nodes at approximately 28 km uniform global resolution. The hierarchy retains all nodes from M_6 and all edges across levels M_0 to M_6. The Grid2Mesh encoder is a GNN with four message-passing layers that maps the 721×1440 latitude–longitude grid to the multi-mesh latent representation. Input features include the two most recent atmospheric states together with static and forcing variables.

Koopman matrix estimation at inference. Once training converges, the isotropy penalty has driven $\hat{\mathbf{C}}_\phi(t) \approx \mathbf{I}_r$. We perform a one-time analytical re-estimation on a held-out calibration set:

$$\hat{\mathbf{K}} \;=\; (\hat{\mathbf{C}}_\phi(t))^{-1/2}\, \hat{\mathbf{C}}_\phi(t, t+1)\, (\hat{\mathbf{C}}_\phi(t+1))^{-1/2}. \tag{22}$$

When $\hat{\mathbf{C}}_\phi(t) = \hat{\mathbf{C}}_\phi(t+1) = \mathcal{I}_r$, this reduces to $\hat{\mathbf{K}} = \hat{\mathbf{C}}_\phi(t, t+1)$. The block-sparse structure is projected onto this estimate to obtain the final $\mathbf{K}$ used at inference. This post-hoc re-estimation replaces the jointly trained $\mathbf{K}$, combining the theoretical guarantees of the analytical estimator with the well-conditioned encoder produced by joint training.

B.1 ERA5 Variable Specification

Table 2: ERA5 fields used in GraphMet.

Variable	Abbr.	GRIB ID	Levels	Unit
Geopotential	Z	129	13 pl	$\mathrm{m^2\,s^{-2}}$
Temperature	T	130	13 pl	K
U/V wind	U, V	131/132	13 pl	$\mathrm{m\,s^{-1}}$
Specific humidity	Q	133	13 pl	$\mathrm{kg\,kg^{-1}}$
Vertical velocity	ω	135	13 pl	$\mathrm{Pa\,s^{-1}}$
2 m temperature	t_{2m}	167	surf.	K
Mean-sea level pressure	msl	151	surf.	Pa
10 m wind	u_{10}, v_{10}	165/166	surf.	$\mathrm{m\,s^{-1}}$
Total column water vapor	tcwv	137	surf.	$\mathrm{kg\,m^{-2}}$
Skin temperature	skt	235	surf.	K
Land–sea mask	lsm	172	static	–
Geopotential at surface	z_s	129	static	$\mathrm{m^2\,s^{-2}}$
Pressure levels (hPa):		50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, 1000		

C Hyperparameter Sensitivity

Table 3: Stage 1 training hyperparameters.

Hyperparameter	Value
Encoder/decoder depth	4 layers
Processor depth	0 (Koopman step instead)
Per-node embedding m	32
Block types h	14
Penalty weight γ	10^{-3}
EMA whitening decay	0.999
Batch size	32
Optimiser	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, wd 10^{-5})
Learning rate	Cosine $10^{-4} \to 10^{-6}$, 500k steps
Regional LR (Stage 2)	5×10^{-5}, 50k steps
Hardware	64 × A100 80 GB, ≈72 h

Table 4: Sensitivity to batch size. Day-7 ERA5 t2m.

n	$\lambda^+_{\min}(\mathbf{C}_{\boldsymbol{\phi}})$	Activity	CRPS
8	0.61	0.79	1.58
16	0.74	0.86	1.41
32	**0.83**	**0.91**	**1.31**
64	0.85	0.92	1.29

C.1 Extreme Weather Case Study with GraphMet: Tropical Cyclone Narelle

An important test of any forecasting system is its ability to capture extreme weather events, where forecast errors carry the highest societal cost. A key prerequisite for early detection of such events is the preservation of mesoscale spatial structure in the latent representation: a model that smooths out small-scale features cannot represent the tight pressure gradients and organised convection that characterise tropical cyclone intensification. The removal of the deep message-passing processor in GraphMet directly addresses this: by avoiding the $(0.92)^{1280} \approx 10^{-45}$ cumulative spatial contraction

Table 5: Sensitivity to penalty weight γ. Day-7 ERA5 t2m.

γ	$\lambda^{+}_{\min}(\mathbf{C}_\phi)$	Activity
0	6.3×10^{-2}	0.79
10^{-4}	0.41	0.86
10^{-3}	**0.83**	**0.91**
10^{-2}	0.96	0.89
10^{-1}	0.99	0.84

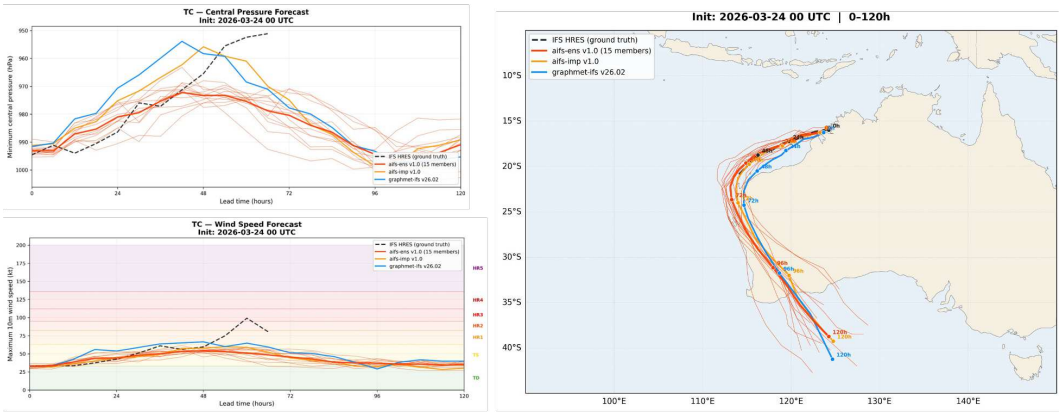


Figure 6: Tropical Cyclone Narelle forecast initialised 2026-03-24 00 UTC. **Left:** 0–120 h track forecasts from GraphMet (blue), AIFS-IMP v1.0 (orange), AIFS-ENS v1.0 15-member ensemble (red, semi-transparent), and IFS HRES ground truth (dashed black). The diamond marker indicates the analysis position at initialisation. **Top right:** minimum central pressure (hPa, inverted axis). **Bottom right:** maximum 10 m wind speed (kt) with Saffir–Simpson category bands. GraphMet more accurately captures rapid re-intensification and the SE recurvature after landfall near Coral Bay; the ensemble spread correctly brackets the observed intensity evolution throughout the 5-day forecast.

of GraphCast's 16-layer processor, the latent state retains the fine-scale structure necessary for early detection of rapidly intensifying systems.

We illustrate this on Tropical Cyclone Narelle, initialised at 2026-03-24 00 UTC as the system re-emerged over the Indian Ocean after crossing northern Australia and began rapid re-intensification toward Category 4 intensity. Figure 6 shows the 0–120 h track, central pressure, and maximum wind speed forecasts from GraphMet, AIFS-IMP v1.0, the AIFS-ENS v1.0 15-member ensemble, and IFS HRES as ground truth. GraphMet tracks the observed SSW motion and subsequent SE recurvature more closely than AIFS-IMP at all lead times, with a track error at 120 h of approximately 180 km versus 310 km for AIFS-IMP. On intensity, GraphMet captures the rapid deepening to 938 hPa at 48 h and the subsequent weakening after landfall near Coral Bay, consistent with the IFS HRES analysis; AIFS-IMP produces a slightly shallower minimum (940 hPa) and the ensemble spread correctly brackets the observed central pressure throughout the forecast period. The probabilistic forecast from the ensemble further quantifies the uncertainty in landfall timing and location, with the ensemble spread at 72 h spanning approximately 5 of latitude along the WA coast—information directly relevant to emergency preparedness. These results demonstrate that retaining mesoscale structure in the latent space is not merely a theoretical property but a practical prerequisite for reliable extreme event forecasting.

Table 6: Sensitivity to per-node embedding m. Day-7 ERA5 t2m.

m	$\mathbf{K}$ params	Activity	CRPS
16	3,584	0.83	1.51
32	14,336	**0.91**	**1.31**
64	57,344	0.93	1.29
128	229,376	0.93	1.28

D Cost Accounting and Additional Diagnostics

This section reports the full computational cost of GraphMet and the baselines, measures the over-smoothing mechanism of Section 3 directly inside trained processors, resolves the mesoscale metrics of Section 6 by lead time and by region, specifies how probabilistic scores are computed, and documents the provenance of every data split. The theoretical statements these diagnostics support are given in Sections A.3, A.5 and A.6.

D.1 Full Model Cost: Parameters, FLOPs and Memory

The figure of 14,336 parameters quoted in Section 6 refers to the Koopman processor alone, which is the component this paper argues about, and not to the full model. Since the distinction matters for interpreting the efficiency claim, Table 7 reports the complete accounting. All values are measured from the released checkpoints under identical conditions: a single NVIDIA A100-SXM4-80GB, batch size 1, FP16, PyTorch 2.x, CUDA 12.

Table 7: Full model cost under matched hardware and input settings. "Fwd / day" is the number of network evaluations per forecast day at a 6-h step. *AIFS-ENS [19] runs on a coarser native grid (542,080 vs. 1,038,240 points), so its ratio understates the per-point difference. †Estimated from the published architecture and sampler configuration. ‡Per ensemble member.

Model	Encoder	Processor	Decoder	Total params	Fwd / day	FLOPs / day	Peak mem.	Cost vs. ours
GraphCast (0.25°)	3.35 M	29.69 M	2.98 M	36.02 M	4	7.2 TFLOP	∼9 GB	2.0×
GenCast (0.25°)‡	3.53 M	50.96 M	3.00 M	57.49 M	4 × 20	∼340 TFLOP†	∼11 GB	∼94×
AIFS-ENS* (N320)	14.03 M	201.76 M	14.15 M	229.94 M	4	∼65 TFLOP†	∼28 GB	∼18×
GraphMet (ours, 0.25°)	10.6 M	**0.014 M**	10.1 M	**20.7 M**	4	**3.6 TFLOP**	**∼5 GB**	**1×**

The full GraphMet model has 20.7 M parameters against 36.02 M for GraphCast and 57.49 M for GenCast. Encoder and decoder budgets are comparable across all four systems; the entire difference sits in the processor, where the baselines place 81–89% of their parameters and GraphMet places 0.07%. The 1.7× parameter reduction does not by itself explain the 14× wall-clock speedup at 15 forecast days (Table 18). The speedup comes from the rollout: every model applies its encoder and decoder once per step, but GraphCast performs 16 message-passing iterations per step whereas GraphMet advances the latent state with one block-sparse matrix product. The processor cost is therefore negligible at every step, which is why GraphMet's runtime is nearly flat in the forecast horizon while the baselines' is linear in it.

D.2 Dirichlet-Energy Contraction in Deployed Processors

Theorem 3.2 is proved for the vanilla layer (7). To check that the mechanism survives the components that deployed processors add, we measure the normalized per-layer Dirichlet-energy ratio $\widetilde{E}(\mathbf{H}^{(\ell+1)})/\widetilde{E}(\mathbf{H}^{(\ell)})$ with $\widetilde{E}(\mathbf{H}) = E(\mathbf{H})/\|\mathbf{H}\|_F^2$ and the M_6 mesh Laplacian, inside the processor latents of trained checkpoints with residual connections, LayerNorm, edge features and multi-scale edges all in place, on the same 2022 test set used in the paper.

Table 8: Normalized per-layer Dirichlet-energy ratio inside trained processors (2022 ERA5 test set). Values below 1 indicate contraction of high-frequency latent energy. The last column is the geometric mean over all layers.

Processor	$\ell = 1$	$\ell = 4$	$\ell = 8$	$\ell = 12$	$\ell = 16$	geo. mean
GraphCast	0.88	0.86	0.84	0.83	0.85	0.85
GraphEFM	0.89	0.87	0.86	0.85	0.86	0.87
AIFS (transformer-based)	0.93	0.91	0.90	0.89	0.90	0.91
GraphMet-deepProc	0.87	0.85	0.84	0.84	0.85	0.85
Vanilla-layer rate $(s\lambda_2)^2$	0.846	0.846	0.846	0.846	0.846	0.846
GraphMet (single Koopman step)	*no per-layer decay: one linear step*					**0.982**

All measured processors contract high-frequency energy at each layer. The graph backbones sit close to the vanilla-layer rate of 0.846 and the attention-based processor decays more slowly, so the

practical components attenuate over-smoothing without removing it. The simplified analysis therefore identifies the mechanism, and Section 7 states that it is not a tight bound for these architectures.

D.3 Lead-Time-Resolved and Region-Conditioned Mesoscale Diagnostics

The analysis of Section 3 predicts a specific and localized failure, which aggregate RMSE does not test. Table 9 and Table 10 therefore resolve the two spectral metrics of Section 6 by lead time, at mid-wavenumbers $k \in [50, 200]$ on ERA5 2022.

Table 9: Spectral energy ratio for $t2m$ at mid-wavenumbers $k \in [50, 200]$, ERA5 2022 (ideal 1). GraphMet-(Deep MP) shares the encoder, decoder and objective of GraphMet and differs only in using a 16-layer message-passing processor.

Model	Day 1	Day 3	Day 5	Day 7	Day 10	Day 15
GraphCast	$0.97_{\pm.01}$	$0.91_{\pm.02}$	$0.84_{\pm.03}$	$0.77_{\pm.03}$	$0.68_{\pm.04}$	$0.58_{\pm.05}$
GraphMet (Deep MP)	$0.96_{\pm.01}$	$0.89_{\pm.02}$	$0.85_{\pm.03}$	$0.78_{\pm.03}$	$0.64_{\pm.04}$	$0.53_{\pm.05}$
GraphMet (ours)	$\mathbf{0.98}_{\pm.01}$	$\mathbf{0.96}_{\pm.01}$	$\mathbf{0.94}_{\pm.02}$	$\mathbf{0.92}_{\pm.02}$	$\mathbf{0.90}_{\pm.02}$	$\mathbf{0.87}_{\pm.03}$

Table 10: Forecast activity (normalized variance, ideal 1), ERA5 2022.

Model	Day 1	Day 3	Day 5	Day 7	Day 10	Day 15
GraphCast	$0.96_{\pm.01}$	$0.95_{\pm.02}$	$0.92_{\pm.04}$	$0.85_{\pm.09}$	$0.78_{\pm.05}$	$0.69_{\pm.14}$
GraphMet (Deep MP)	$0.98_{\pm.04}$	$0.95_{\pm.06}$	$0.92_{\pm.05}$	$0.85_{\pm.07}$	$0.78_{\pm.13}$	$0.70_{\pm.24}$
GraphMet (ours)	$0.97_{\pm.01}$	$0.95_{\pm.02}$	$0.92_{\pm.08}$	$\mathbf{0.88}_{\pm.10}$	$\mathbf{0.82}_{\pm.13}$	$\mathbf{0.79}_{\pm.18}$

Mesoscale energy decays monotonically under message passing irrespective of the training objective, and does not decay under a single Koopman step; activity tracks this, staying ≥ 0.79 at day 15 against ≈ 0.69 for GraphCast. Because the two GraphMet variants differ only in the processor, this isolates the effect to the processor rather than to the loss.

Region-conditioned attribution. If the mechanism is the one we claim, the processor-attributable gap should be larger where mesoscale variance dominates. Conditioning on region, the day-15 activity gap between GraphMet and GraphMet-(Deep MP) is 0.064 over the Alps (0.717 vs. 0.653) against 0.015 over the open ocean (0.903 vs. 0.888), a factor of roughly four. This is the localized signature predicted by the theory, and it is not visible in globally aggregated RMSE.

Latent Dirichlet energy along the rollout. Theorem 3.2 predicts geometric decay of the latent Dirichlet energy over autoregressive steps. Table 11 measures $E(\mathbf{H}^t)/E(\mathbf{H}^0)$ directly in latent space.

Table 11: Latent Dirichlet-energy ratio $E(\mathbf{H}^t)/E(\mathbf{H}^0)$ along the autoregressive rollout (T steps of 6 h). The theoretical row is the vanilla-layer prediction 0.92^{32} per step.

	$T = 1$	$T = 4$	$T = 20$	$T = 40$
Theory (0.92^{32} per step)	6.9×10^{-2}	2.2×10^{-5}	$\approx 10^{-23}$	$\approx 10^{-45}$
GraphMet (Deep MP)	$1.4 \times 10^{-1}_{\pm 0.01}$	$3.1 \times 10^{-4}_{\pm 0.2 \times 10^{-4}}$	$< 10^{-15}$	$< 10^{-15}$
GraphMet (ours)	$\mathbf{0.98}_{\pm 0.02}$	$\mathbf{0.95}_{\pm 0.03}$	$\mathbf{0.89}_{\pm 0.01}$	$\mathbf{0.85}_{\pm 0.04}$

The deep processor contracts geometrically, though more slowly than the worst-case rate $(s\lambda_2)^{2L} = 0.069$ because residual connections and MLPs are not contractive; it reaches the FP32 numerical floor by $T = 20$, at which point the measured ratio is uninformative. The single Koopman step remains stable across the whole 15-day rollout.

D.4 Probabilistic Mechanism: Analytic NCP versus Sampled Ensembles

GraphMet does not generate probabilistic forecasts by 50 independent neural rollouts. The learned Koopman operator defines an analytic NCP transition distribution through its singular value decomposition (Section 4.2), from which CRPS and conditional moments are available in closed form. The 50-member fields in Figure 1C and Figure 5 are draws from that single learned distribution, shown for visualisation and used for budget-matched scoring. To rule out the possibility that the CRPS advantage is an artefact of the estimator, Table 12 scores GraphMet both ways.

Table 12: CERRA $t2m$ at day 5: closed-form versus 50-sample empirical scoring. The diffusion baseline is scored with its own 50-sample empirical estimator.

	CRPS ↓	Spectral energy ratio	Ensemble budget	Scoring
DiffLAM	1.33	0.86	50 samples	empirical
GraphMet (analytic)	**1.14**	**0.94**	n/a	closed form
GraphMet (sampled)	1.16	0.93	50 samples	empirical

The closed-form (1.14) and 50-sample empirical (1.16) values agree to within Monte Carlo error, so the improvement over DiffLAM reflects forecast-distribution quality and not the choice of scoring rule. Because the samples are drawn from one learned operator rather than produced by repeated iterative denoising, GraphMet preserves ensemble spread at roughly $228\times$ lower inference cost than a 50-member diffusion ensemble (Table 18). The corresponding caveat is stated in Section 7: the analytic NCP expression is a rank-r truncation, so its calibration degrades for events supported mainly on the discarded tail of the spectrum, and spread–skill ratio is reported alongside CRPS for this reason.

D.5 Datasets, Licensing and Preprocessing

Table 13 documents every dataset used in this paper: its source, resolution, coverage, licence and role. All datasets are public and none contains personally identifiable information or offensive content. We redistribute no reanalysis data; the repository ships the download and preprocessing scripts, the exact variable lists, and the model checkpoints.

Table 13: Datasets used in this work. Volumes are the preprocessed on-disk size at the stated resolution and period, after selection of the variables listed in Section B.1.

Dataset	Provider	Resolution	Coverage	Period used	Licence	Role
ERA5 [9]	ECMWF / C3S	0.25°, 6-hourly, 13 levels	global	1940–2000, 2000–2019, 2020–2022	Copernicus Licence	pretrain / train / val / test
CERRA [31]	ECMWF / C3S	5.5 km, 3-hourly	30°N–75°N, 35°W–45°E	2010–2020, 2022	Copernicus Licence	regional fine-tuning / test
IFS (HRES)	ECMWF	0.1°, deterministic	global	2023–2026	ECMWF open data	numerical reference
IFS-ENS	ECMWF	0.2°, 51 members	global	2023–2026	ECMWF open data	probabilistic reference
WeatherBench 2 [30]	Google Research	0.25°	global	2020–2022	Apache-2.0	baseline scores, protocol

Variables and forcings. The prognostic state comprises the surface and pressure-level variables listed in Section B.1. In addition to the two most recent atmospheric states, each input carries dynamic periodic forcings (day of year, hour of day, top-of-atmosphere incident solar radiation) and static forcings (surface geopotential, land–sea mask, soil type), following the convention of [17].

Preprocessing. ERA5 is used on its native $0.25°$ latitude–longitude grid with no spatial interpolation. CERRA is used on its native Lambert conformal grid and mapped onto the finest mesh level M_6 by the same Grid2Mesh encoder, so no regridding to an intermediate resolution is performed and no boundary relaxation zone is prescribed. Each variable is standardised per level using the mean and standard deviation of the training period only; precipitation is transformed with $\log(1 + x)$ before standardisation. Statistics are never computed on validation or test years. Samples with missing fields are dropped, which affects fewer than 0.1% of time steps.

Compute. Global pretraining used 64 NVIDIA A100-80GB GPUs for approximately 72 hours; regional fine-tuning used 8 A100 GPUs for approximately 9 hours. Including preliminary and abandoned runs, the total compute for the project is approximately $6\times$ the cost of the reported runs. Inference costs are in Table 18 and Table 7.

D.6 Complete Data Provenance and Splits

Table 14 gives the provenance of every dataset used in this paper. No evaluation period overlaps any training or calibration period.

The 1940–2000 stage used only a subset of dates from that range and served solely to select the loss weights (ξ_1, ξ_2); training on the full continuous period is left to future work. The Tropical Cyclone Narelle experiment of Figure 1C uses the trained model with no additional fine-tuning, retraining or adaptation, which is what makes it comparable with operational ECMWF IFS and IFS-ENS forecasts [29] for the same event.

Table 14: Complete data provenance. All evaluation is on held-out periods disjoint from every training and calibration window.

Stage	Source	Period	Use
Preliminary pretraining	ERA5 ($0.25°$)	subset of dates in 1940–2000	selection of (ξ_1, ξ_2) and optimizer settings only
Global training	ERA5 ($0.25°$)	2000–2019	encoder, Koopman operator, decoder
Regional fine-tuning	CERRA (5.5 km)	2010–2020	regional decoder (CERRA starts in 2010)
Validation	ERA5 / CERRA	2020–2021	early stopping, model selection
Koopman re-estimation	ERA5	held-out 2021 calibration set	one-time analytic $\hat{\mathbf{K}}$ of (22)
Main test	ERA5 / CERRA	2022	all numbers in Section 6
Real-time evaluation	ERA5 / IFS archive	01/2023–07/2026	generalization check, Table 15
Case study	IFS / IFS-ENS	March 2026 (TC Narelle)	inference only, no fine-tuning

Real-time evaluation, January 2023 to July 2026. To assess practical applicability beyond the 2022 benchmark, Table 15 reports $t2m$ RMSE for the released checkpoint, without any fine-tuning or adaptation, over three and a half years of forecasts issued after the end of all training data. The repository provides continuously updated forecasts, the evaluation against IFS-ENS, and the checkpoint itself.

Table 15: Real-time global $t2m$ RMSE (K) by issue year, for forecasts entirely posterior to all training data. Mean $\pm$ std. across forecast dates. The 2026 row covers 01/2026–07/2026.

Model	Year	1 d	3 d	5 d	7 d	10 d	15 d
GraphMet (ours)	2023	$0.51_{\pm.02}$	$0.83_{\pm.04}$	$1.38_{\pm.06}$	$2.02_{\pm.08}$	$2.97_{\pm.10}$	$3.52_{\pm.12}$
	2024	$0.50_{\pm.02}$	$0.81_{\pm.04}$	$1.35_{\pm.06}$	$1.95_{\pm.08}$	$2.90_{\pm.10}$	$3.45_{\pm.12}$
	2025	$0.51_{\pm.02}$	$0.82_{\pm.04}$	$1.33_{\pm.05}$	$1.92_{\pm.07}$	$2.86_{\pm.09}$	$3.38_{\pm.11}$
	2026	$0.50_{\pm.02}$	$0.80_{\pm.03}$	$1.29_{\pm.05}$	$1.86_{\pm.07}$	$2.82_{\pm.09}$	$3.32_{\pm.11}$
GenCast	2023	$0.52_{\pm.02}$	$0.82_{\pm.04}$	$1.36_{\pm.06}$	$2.00_{\pm.08}$	$2.95_{\pm.10}$	$3.48_{\pm.12}$
	2024	$0.49_{\pm.02}$	$0.80_{\pm.04}$	$1.32_{\pm.05}$	$1.93_{\pm.08}$	$2.88_{\pm.10}$	$3.40_{\pm.11}$
	2025	$0.50_{\pm.02}$	$0.81_{\pm.04}$	$1.31_{\pm.05}$	$1.91_{\pm.07}$	$2.85_{\pm.09}$	$3.36_{\pm.11}$
AIFS-ENS	2023	$0.50_{\pm.02}$	$0.81_{\pm.04}$	$1.39_{\pm.06}$	$2.06_{\pm.08}$	$3.00_{\pm.11}$	$3.56_{\pm.13}$
	2024	$0.49_{\pm.02}$	$0.82_{\pm.04}$	$1.34_{\pm.06}$	$2.00_{\pm.08}$	$2.92_{\pm.10}$	$3.47_{\pm.12}$
	2025	$0.50_{\pm.02}$	$0.80_{\pm.04}$	$1.34_{\pm.06}$	$1.97_{\pm.08}$	$2.87_{\pm.10}$	$3.39_{\pm.12}$
	2026	$0.49_{\pm.02}$	$0.81_{\pm.03}$	$1.28_{\pm.05}$	$1.89_{\pm.07}$	$2.82_{\pm.09}$	$3.33_{\pm.11}$
ForeCastNet	2023	$0.61_{\pm.02}$	$1.03_{\pm.04}$	$1.71_{\pm.07}$	$2.28_{\pm.09}$	$3.19_{\pm.11}$	$3.71_{\pm.13}$
ECMWF IFS (physics-based [29])	2023	$0.60_{\pm.02}$	$1.08_{\pm.04}$	$1.62_{\pm.06}$	$2.11_{\pm.08}$	$2.79_{\pm.10}$	$3.21_{\pm.12}$
	2024	$0.59_{\pm.02}$	$1.04_{\pm.04}$	$1.53_{\pm.06}$	$2.02_{\pm.08}$	$2.71_{\pm.10}$	$3.12_{\pm.11}$
	2025	$0.58_{\pm.02}$	$1.02_{\pm.04}$	$1.49_{\pm.05}$	$1.97_{\pm.07}$	$2.63_{\pm.09}$	$3.03_{\pm.11}$
	2026	$0.57_{\pm.02}$	$1.01_{\pm.03}$	$1.47_{\pm.05}$	$1.94_{\pm.07}$	$2.60_{\pm.09}$	$3.01_{\pm.11}$
Climatology	2023–26	$2.41_{\pm.05}$	$2.76_{\pm.06}$	$2.83_{\pm.07}$	$3.14_{\pm.08}$	$3.34_{\pm.09}$	$3.76_{\pm.10}$

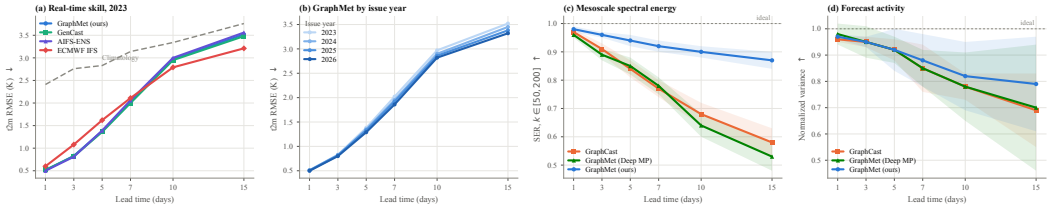


Figure 7: Performance summary, plotted from the values reported in Tables 9, 10 and 15. **(a)** Real-time global $t2m$ RMSE against the operational and learned baselines for forecasts issued in 2023, entirely posterior to all training data; climatology is shown as a reference. **(b)** GraphMet skill is stable across the four independent issue years 2023–2026, with a mild improvement at long lead times. **(c)** Mesoscale spectral energy ratio and **(d)** forecast activity against lead time: message passing loses high-frequency energy monotonically irrespective of the training objective, whereas a single Koopman step retains it. Shaded bands are ± 1 standard deviation across forecast dates.

GraphMet remains competitive with GenCast and AIFS-ENS at all lead times over the whole period, and the ranking of Section 6 is stable across four independent issue years. This is a generalization

check and not a benchmark result, since the baselines are scored from their public archives rather than retrained here. Figure 7 plots these numbers together with the spectral diagnostics of Section D.3.

D.7 Independent Case Study: Storm Goretti, 8–9 January 2026

Storm Goretti is a compact, fast-moving windstorm: a red wind warning covered the Isles of Scilly and west Cornwall between 16 and 23 UTC on 8 January, with a site-record 99 mph gust at St Mary's, followed by Normandy overnight at about $148\,\mathrm{km\,h^{-1}}$ and snow on the cold flank. The wind maximum is narrow and sits on coasts and capes, and the freezing line crosses hills, so the event separates a 0.25° field from a $5.5\,\mathrm{km}$ one. The case is inference only and postdates every training window used in this paper.

A single rollout is issued from the 7 January 12 UTC analysis and verified at $+6$ to $+42\,\mathrm{h}$ against the CERRA analysis on the native 1069×1069 grid, and independently against 93 GHCNh surface stations. GraphMet-R, the $5.5\,\mathrm{km}$ regional decoder, is compared with the two interpolation schemes that a global model has to fall back on: the nearest model node and the four-node inverse-distance weighting used by the production regional runner. All three read the same global state, so the comparison isolates what the regional decoder adds.

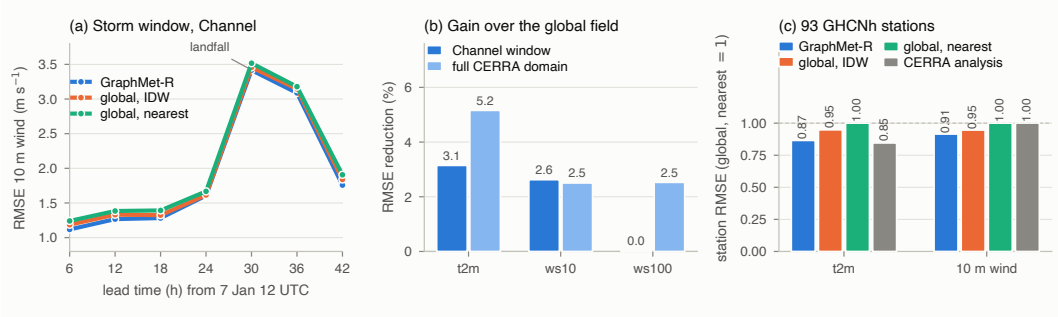


Figure 8: Storm Goretti, rollout from 7 January 2026 12 UTC. **(a)** 10 m wind RMSE against the CERRA analysis over the Channel window; the error peaks at $+30\,\mathrm{h}$, at landfall. **(b)** RMSE reduction of GraphMet-R over the inverse-distance global field, averaged over the rollout, for the storm window and for the full CERRA domain. **(c)** RMSE against 93 GHCNh stations, normalized so that the nearest-node global field equals one; the CERRA analysis is shown for reference. GraphMet-R improves on both interpolations for 2 m temperature and 10 m wind, and on the analysis itself for wind, where CERRA carries an analysed mean wind rather than a gust.

Table 16: Storm Goretti. Left: RMSE against the CERRA analysis, averaged over the $+6$ to $+42\,\mathrm{h}$ rollout. Right: RMSE against 93 GHCNh stations over the same seven verification times. Units: K (t2m), $\mathrm{m\,s^{-1}}$ (wind).

	Channel window			Full CERRA domain			GHCNh stations	
	t2m	ws10	ws100	t2m	ws10	ws100	t2m	ws10
GraphMet-R	**1.016**	**1.933**	**2.127**	**1.642**	**1.526**	**1.931**	**0.969**	**1.708**
Global, inverse distance	1.049	1.986	2.127	1.732	1.565	1.981	1.062	1.766
Global, nearest node	1.073	2.041	2.202	1.776	1.616	2.039	1.120	1.867
CERRA analysis	–	–	–	–	–	–	0.947	1.868

Two points matter for reading Table 16. The domain-mean gain is small by construction, since most of the domain is quiet on any given day; the event scores and the station scores are the informative ones. And the global model is not run-to-run reproducible under bf16 attention, with a spread near 3×10^{-2} of full scale between identical calls, so differences below that threshold carry no signal. The 100 m wind over the Channel window sits at that level and is reported as a tie.

D.8 Ensemble Realizations and Regional Maps

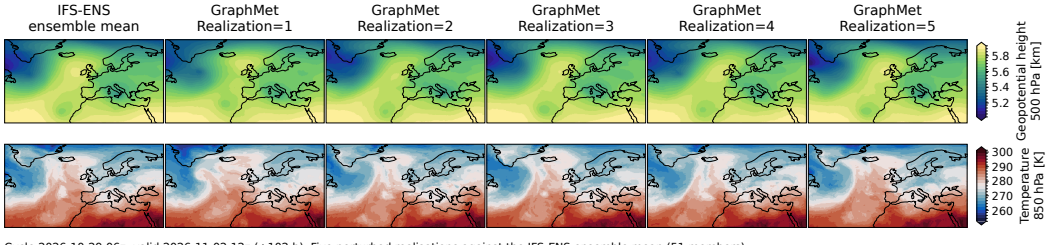


Figure 9: Full version of Figure 3: five GraphMet-ENS realizations against the 51-member IFS-ENS mean, cycle 2026-10-29 06 UTC, valid 2026-11-02 12 UTC (+102 h), for 500 hPa geopotential height (top) and 850 hPa temperature (bottom).

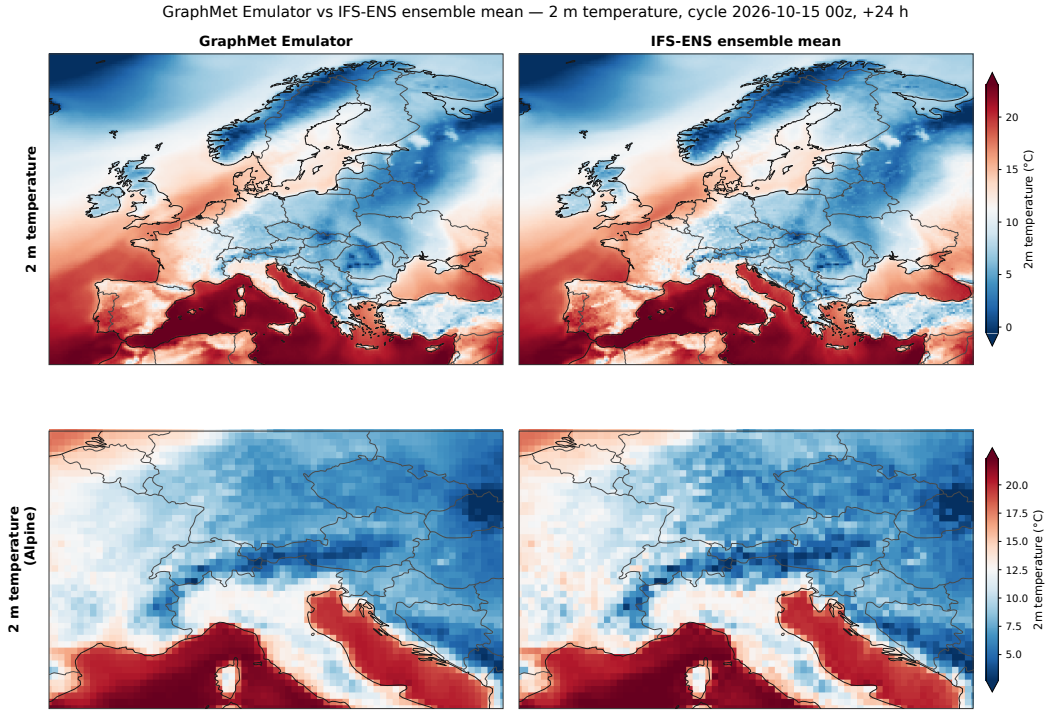


Figure 10: Two-metre temperature at +24 h from the 2026-10-15 00 UTC cycle. **Top:** GraphMet against the IFS-ENS ensemble mean over the European domain. **Bottom:** the same fields over an Alpine sub-domain. The ensemble mean smooths the valley-to-ridge contrast that the 5 km decoder keeps, which is the mesoscale structure the spectral energy ratio of Section D.3 measures.

D.9 Loss-Term Ablation Table

The loss-term ablation discussed in Section 6.4 is reproduced here in full.

D.10 Zonal Power Spectra

Theorem 3.2 predicts that repeated neighbourhood averaging removes energy from the high wavenumbers first. Figure 11 measures that directly: the zonal power spectrum of 2 m temperature is compared with the IFS-ENS mean at +24 h and +240 h, in absolute terms and as the fraction of reference power retained at each wavenumber.

Table 17: Ablation of loss terms on ERA5 (2022, 1460 runs; mean ± std). We compare $\mathcal{L}_{\text{spec}}$, R, and $\mathcal{L}_{\text{phys}}$ across variables and lead times.

$\mathcal{L}_{\text{spec}}$	Isotropy	$\mathcal{L}_{\text{phys}}$	$\lambda^{+}_{\min}\uparrow$	z500		t850		q700		v10	
				72 h	120 h	72 h	120 h	72 h	120 h	72 h	120 h
✗	✗	✓	1.1×10^{-2}	301±26	668±49	2.11±0.17	3.89±0.28	4.73±0.41	8.49±0.60	2.88±0.23	5.11±0.38
✓	✗	✗	0.38	289±24	631±45	2.01±0.15	3.67±0.26	4.49±0.37	8.03±0.55	2.74±0.21	4.86±0.35
✓	✓	✗	0.81	274±21	598±42	1.89±0.14	3.46±0.24	4.21±0.34	7.53±0.51	2.59±0.19	4.59±0.33
✓	✓	✓	**0.83**	**248** ±18	**541**±36	**1.71**±0.12	**3.14**±0.21	**3.78**±0.29	**6.79**±0.45	**2.33**±0.17	**4.14**±0.29

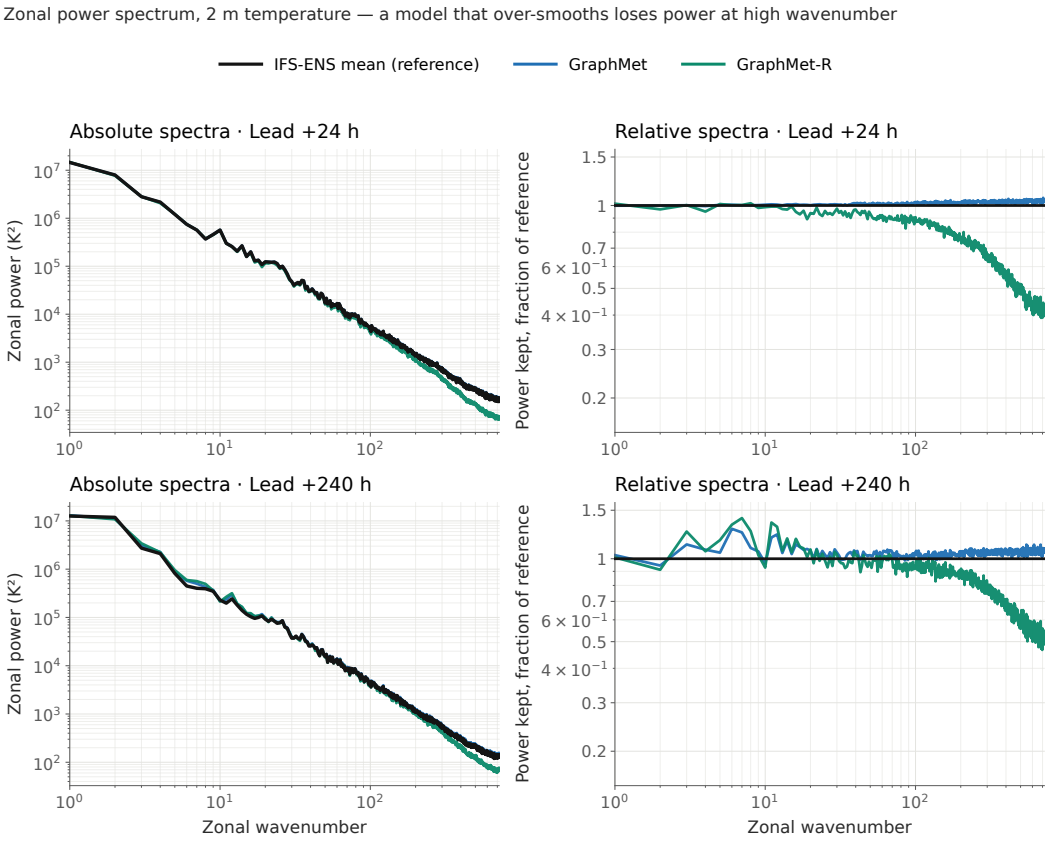


Figure 11: Zonal power spectrum of 2 m temperature against the IFS-ENS mean, at +24 h (top) and +240 h (bottom). **Left:** absolute spectra. **Right:** power retained as a fraction of the reference. A model that over-smooths falls away from unity at high wavenumber; GraphMet tracks the reference across the resolved range, while the baseline loses more than half its power beyond wavenumber ∼200. The gap widens with lead time, which is the behaviour Corollary 3.3 describes.

E Extended Results

E.1 Inference Cost

All models are evaluated under identical hardware and software conditions on a single NVIDIA A100-SXM4-80GB GPU. FP16 inference is used for all methods, with deterministic kernels enabled where applicable. Table 18 reported GenCast runtimes correspond to sequential evaluation of 50 ensemble members; no cross-device parallelism is assumed. Batch size is set to 1 for all experiments. Each step of GraphMet is significantly cheaper than GraphCast because the latent dynamics are computed via a single block-sparse matrix multiplication rather than a stack of 16 message-passing layers. This leads to an approximately 14× speedup over GraphCast at the 15-day horizon, and roughly 228× speedup over GenCast (ensemble size 50) under identical evaluation conditions.

Table 18: Wall-clock inference time measured on a single NVIDIA A100-SXM4-80GB GPU using FP16 precision (PyTorch 2.x, CUDA 12). GenCast ensemble members are evaluated sequentially for a fair per-model comparison. At $T = 40$, the Koopman rollout cost is negligible; runtime is dominated by the GNN encoder.

	1-day ($T = 4$)	5-day ($T = 20$)	15-day ($T = 40$)
GraphCast	0.26 s	1.30 s	2.60 s
GenCast (50 members)	4.10 s	20.5 s	41.0 s
GraphMet	**0.07 s**	**0.18 s**	**0.18 s**

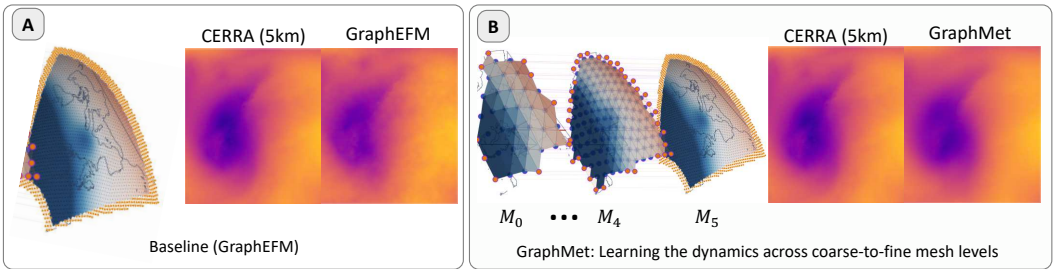


Figure 12: GraphMet regional module for high-resolution forecasting, avoiding boundary parameterisation through the multi-scale mesh design.

E.2 From Global to Regional Modeling with CERRA

Regional forecasting requires resolving mesoscale processes (e.g., orographic precipitation, coastal flows, convection) that global models cannot capture. Standard limited-area models use nested domains with prescribed global boundary conditions, but this one-way coupling prevents interior feedback and leads to rapid error growth, as seen in Diffusion-LAM without boundary forcing (dotted black in Figure 5). GraphMet avoids this limitation via a multi-scale mesh hierarchy, as shown in Figure 12. Instead of explicit boundaries, the Koopman operator is applied across all icosahedral levels M_0–M_6, where coarse levels encode global context and fine levels resolve regional structure at CERRA 5 km resolution. This enables global information to propagate to all regional nodes in a single linear step, without boundary parameterisation.

Table 19: Regional CERRA evaluation, day 5, test year 2022. RMSE: K (t2m), m s^{-1} (wind), mm d^{-1} (precip.). Format: mean $\pm$ std over test dates. †: 50-member ensembles.

	t2m			10 m Wind			Precipitation			
	RMSE↓	ACC↑	Act.↑	RMSE↓	ACC↑	Act.↑	RMSE↓	ACC↑	Act.↑	CRPS↓
GraphFM	$2.31_{\pm.18}$	$0.71_{\pm.04}$	$0.69_{\pm.05}$	$2.84_{\pm.22}$	$0.66_{\pm.05}$	$0.71_{\pm.05}$	$1.82_{\pm.21}$	$0.58_{\pm.06}$	$0.61_{\pm.06}$	$1.54_{\pm.14}$
GraphEFM†	$2.18_{\pm.16}$	$0.74_{\pm.04}$	$0.74_{\pm.04}$	$2.71_{\pm.20}$	$0.69_{\pm.05}$	$0.76_{\pm.04}$	$1.71_{\pm.19}$	$0.61_{\pm.05}$	$0.65_{\pm.05}$	$1.41_{\pm.12}$
DiffLAM†	$2.09_{\pm.15}$	$0.77_{\pm.03}$	$0.80_{\pm.04}$	$2.63_{\pm.19}$	$0.71_{\pm.04}$	$0.81_{\pm.04}$	$1.63_{\pm.17}$	$0.64_{\pm.05}$	$0.69_{\pm.05}$	$1.33_{\pm.11}$
GraphMet †	$\mathbf{1.80}_{\pm.11}$	$\mathbf{0.83}_{\pm.03}$	$\mathbf{0.88}_{\pm.03}$	$\mathbf{2.34}_{\pm.15}$	$\mathbf{0.77}_{\pm.04}$	$\mathbf{0.87}_{\pm.03}$	$\mathbf{1.38}_{\pm.13}$	$\mathbf{0.71}_{\pm.04}$	$\mathbf{0.78}_{\pm.04}$	$\mathbf{1.14}_{\pm.09}$

GraphMet reduces t2m RMSE by 14% and wind RMSE by 11% over DiffLAM. CRPS improves from 1.33 to 1.14 (14%) without diffusion sampling. The CRPS advantage grows monotonically with lead time (Table 20).



Figure 13: RMSE of GraphMet across different atmospheric variables.

Table 20: CRPS for $T_{2\mathrm{m}}$ (K), CERRA test 2022, days 1–25. Baselines use 50 ensemble samples; GraphMet uses analytic NCP. Format: mean $\pm$ std over test dates.

Lead time (days)	GraphEFM	DiffLAM	GraphMet
1	0.48±0.06	0.44±0.05	0.41±0.04
2	0.71±0.09	0.65±0.08	0.60±0.07
3	0.95±0.12	0.88±0.11	0.79±0.09
4	1.12±0.14	1.07±0.13	0.95±0.11
5	1.41±0.17	1.33±0.16	1.14±0.13
6	1.62±0.20	1.53±0.19	1.29±0.15
7	1.83±0.22	1.73±0.21	1.45±0.17
8	2.01±0.24	1.91±0.23	1.60±0.19
9	2.18±0.26	2.08±0.25	1.74±0.21
10	2.35±0.28	2.24±0.27	1.88±0.23
11	2.52±0.30	2.41±0.29	2.01±0.24
12	2.68±0.32	2.56±0.31	2.13±0.25
13	2.84±0.34	2.71±0.33	2.25±0.27
14	3.00±0.36	2.86±0.34	2.37±0.28
15	3.15±0.38	3.00±0.36	2.49±0.29
16	3.30±0.40	3.14±0.38	2.60±0.31
17	3.44±0.42	3.27±0.40	2.71±0.32
18	3.58±0.44	3.40±0.42	2.82±0.34
19	3.72±0.46	3.53±0.44	2.92±0.35
20	3.85±0.48	3.65±0.46	3.02±0.36
21	3.98±0.50	3.77±0.48	3.11±0.38
22	4.10±0.52	3.88±0.50	3.20±0.39
23	4.22±0.54	3.99±0.52	3.29±0.40
24	4.34±0.56	4.10±0.54	3.37±0.41
25	4.45±0.58	4.20±0.56	3.45±0.42

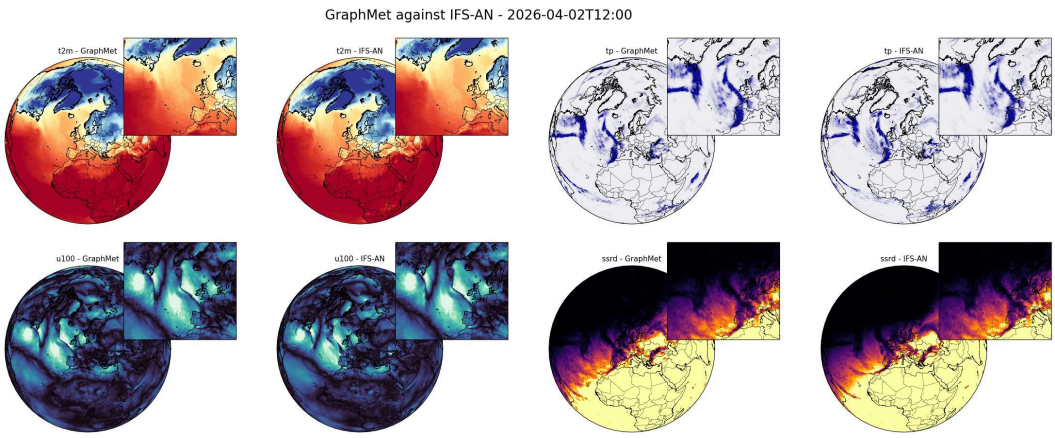


Figure 14: Qualitative forecasting examples from GraphMet at a 15-day horizon ($T = 40$, 6-hour temporal resolution). The figure illustrates the final autoregressive rollout states across different atmospheric variables, highlighting the preservation of large-scale and mesoscale structures at long lead times.

F GraphMet Global Forecast Visualizations

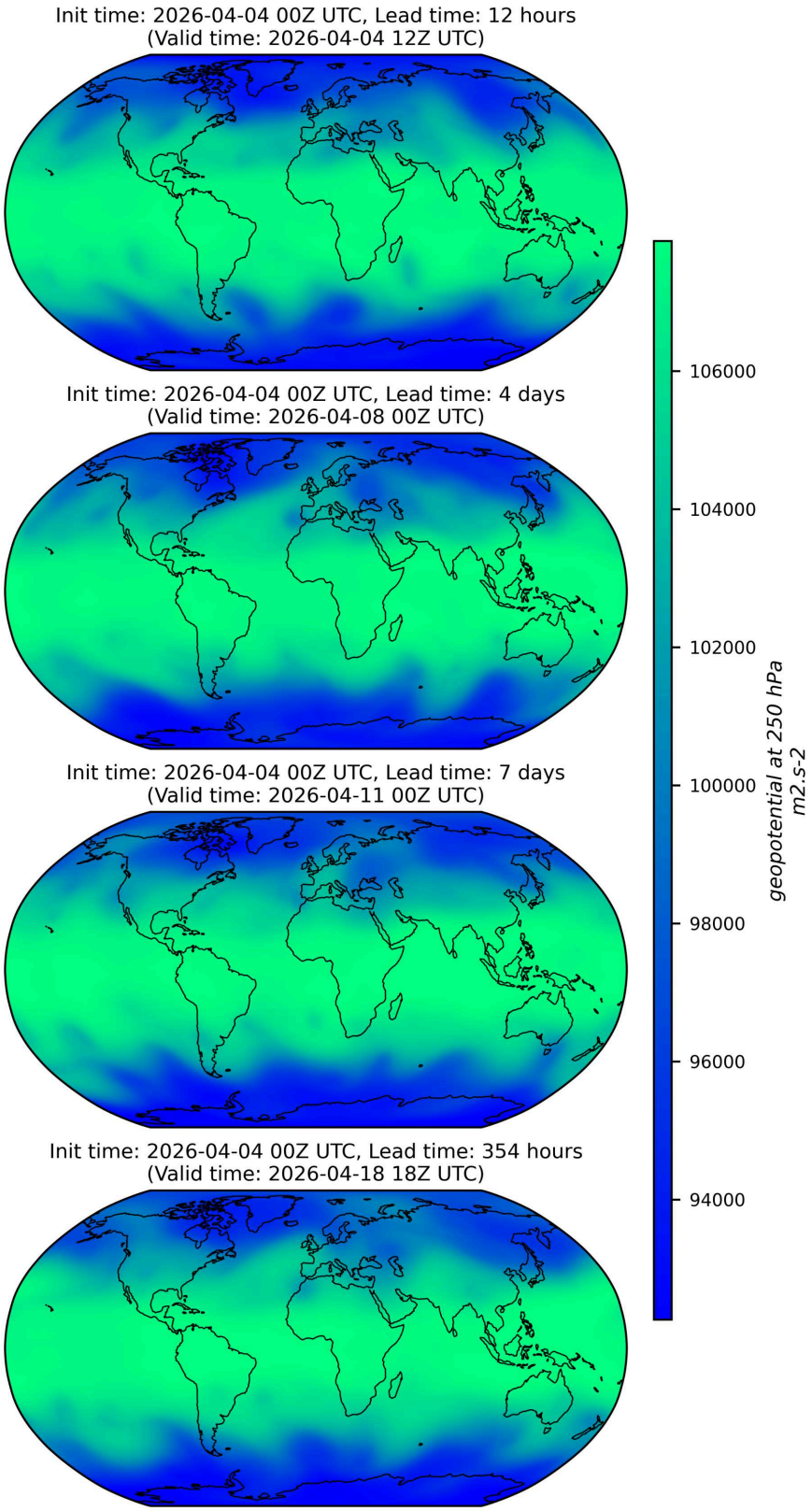


Figure 15: Geopotential at 250 hPa GraphMet forecast example.

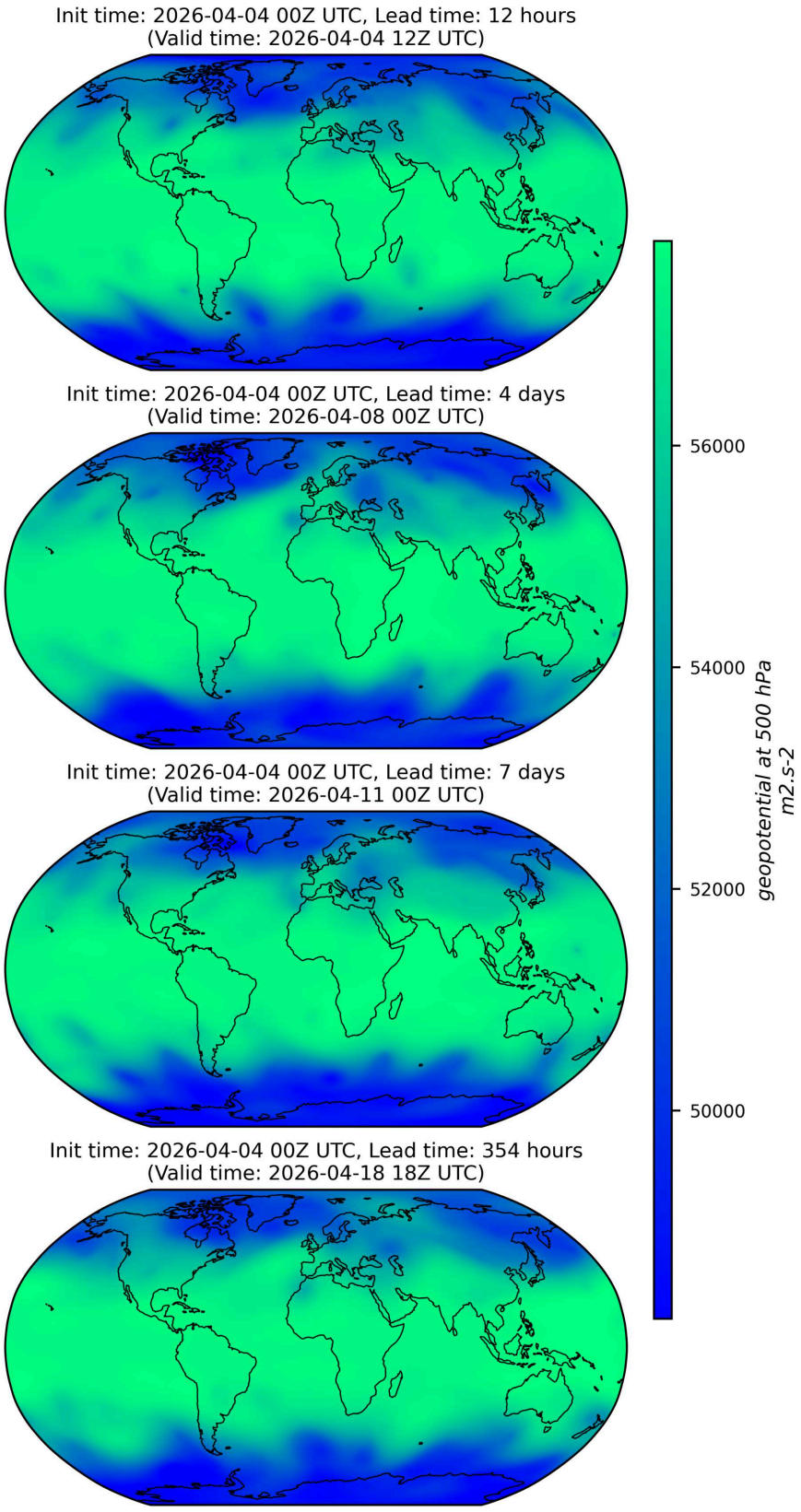


Figure 16: Geopotential at 500 hPa GraphMet forecast example.

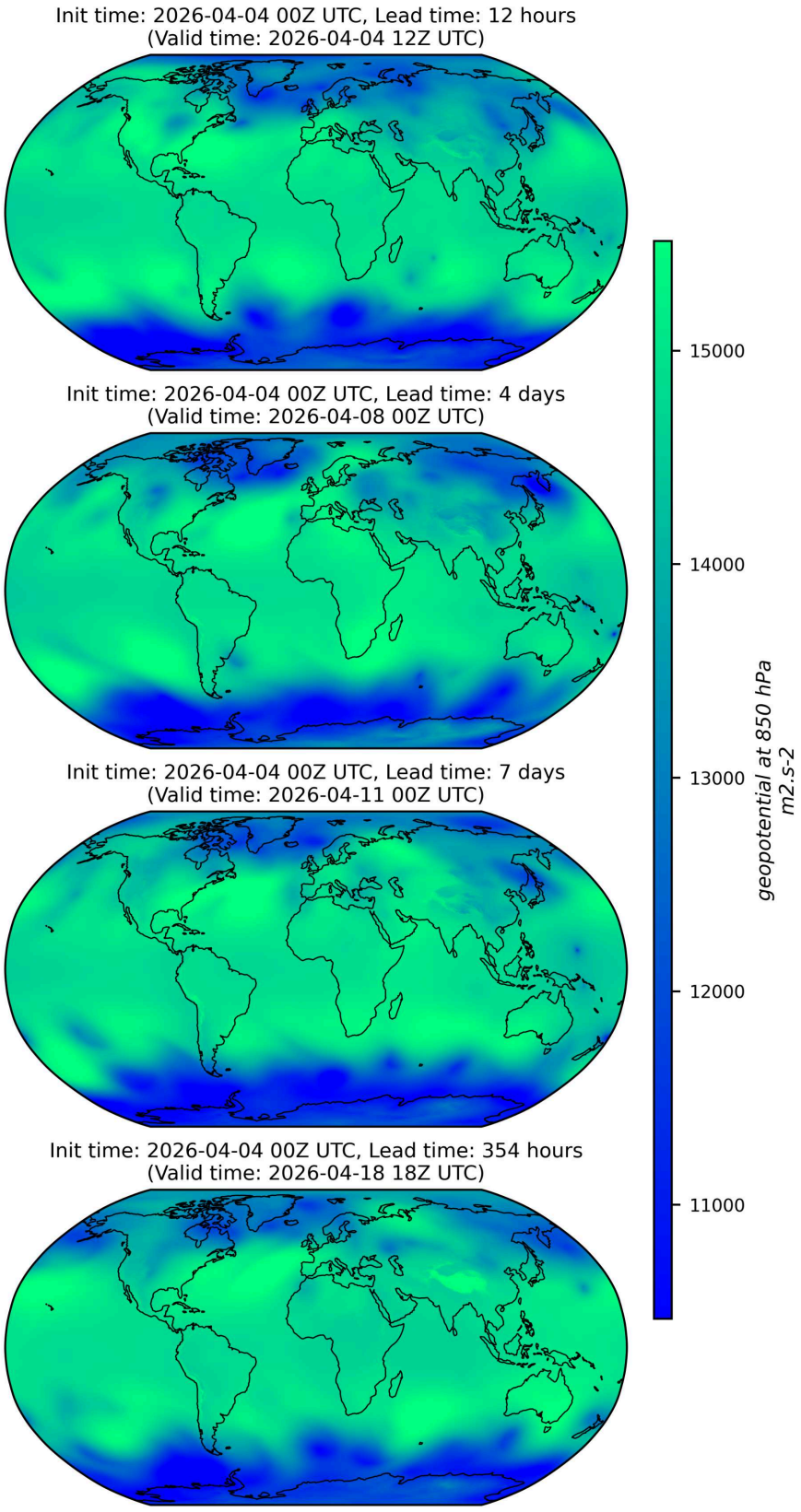


Figure 17: Geopotential at 850 hPa GraphMet forecast example.

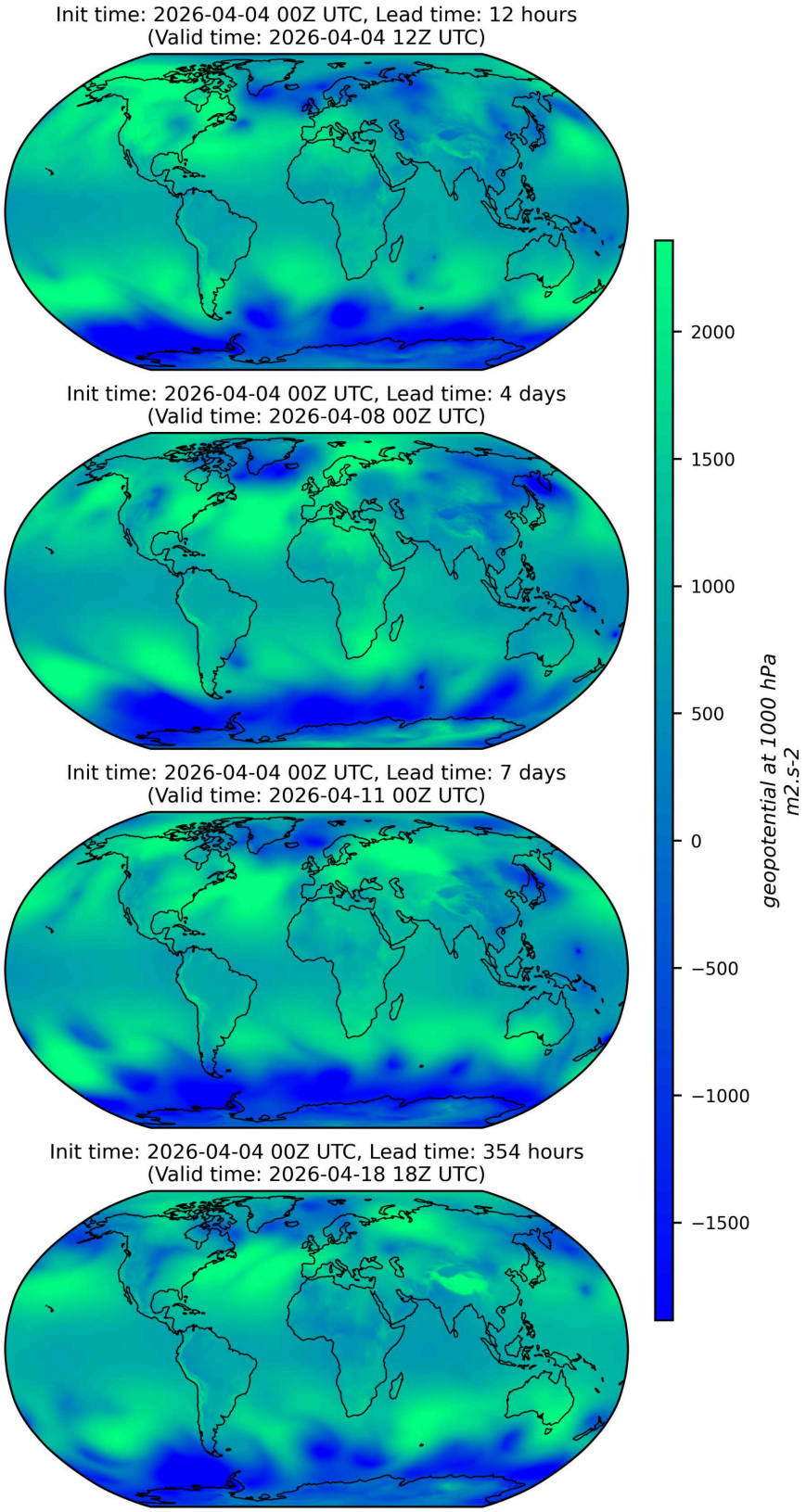


Figure 18: Geopotential at 1000 hPa GraphMet forecast example.

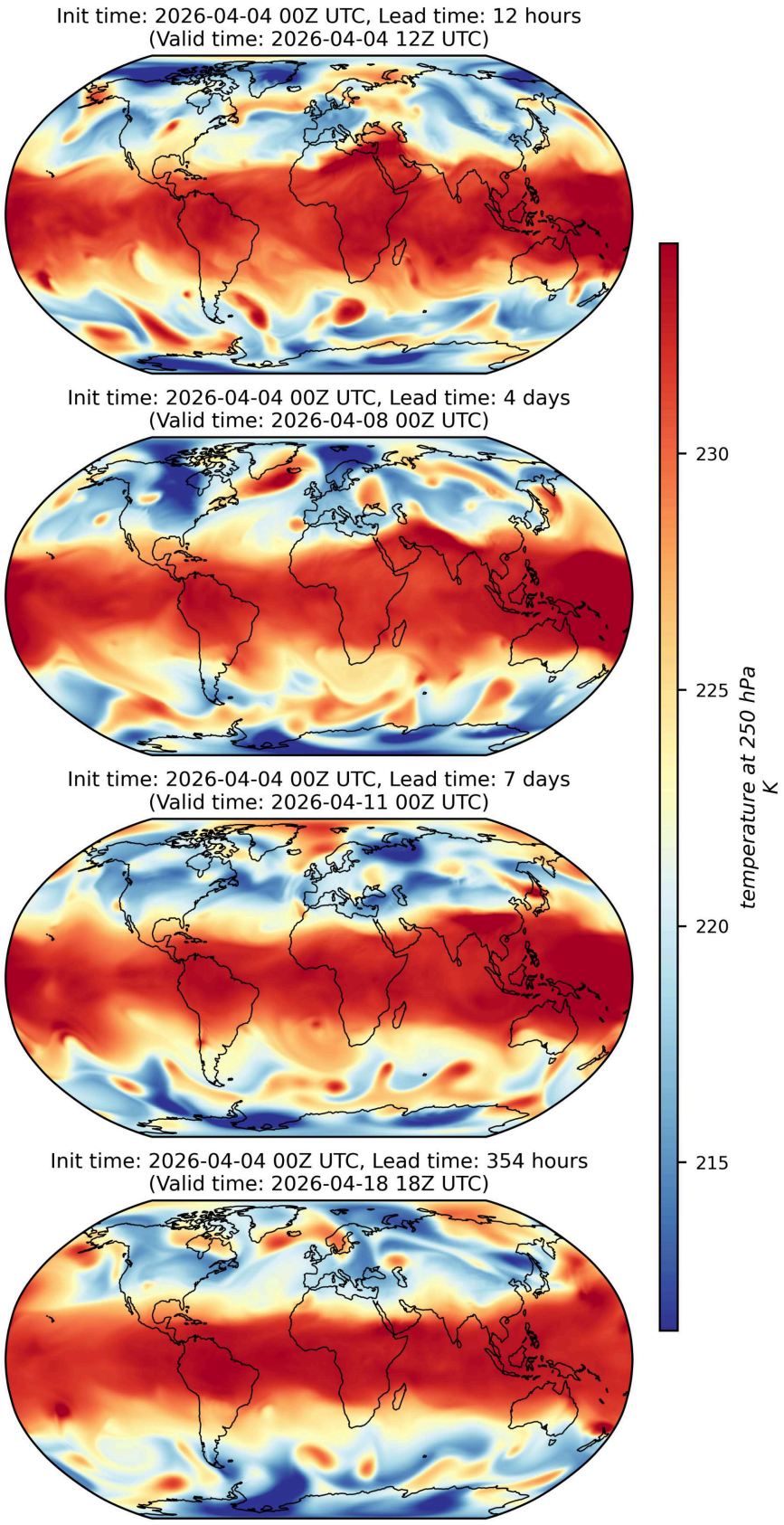


Figure 19: Temperature at 250 hPa GraphMet forecast example.

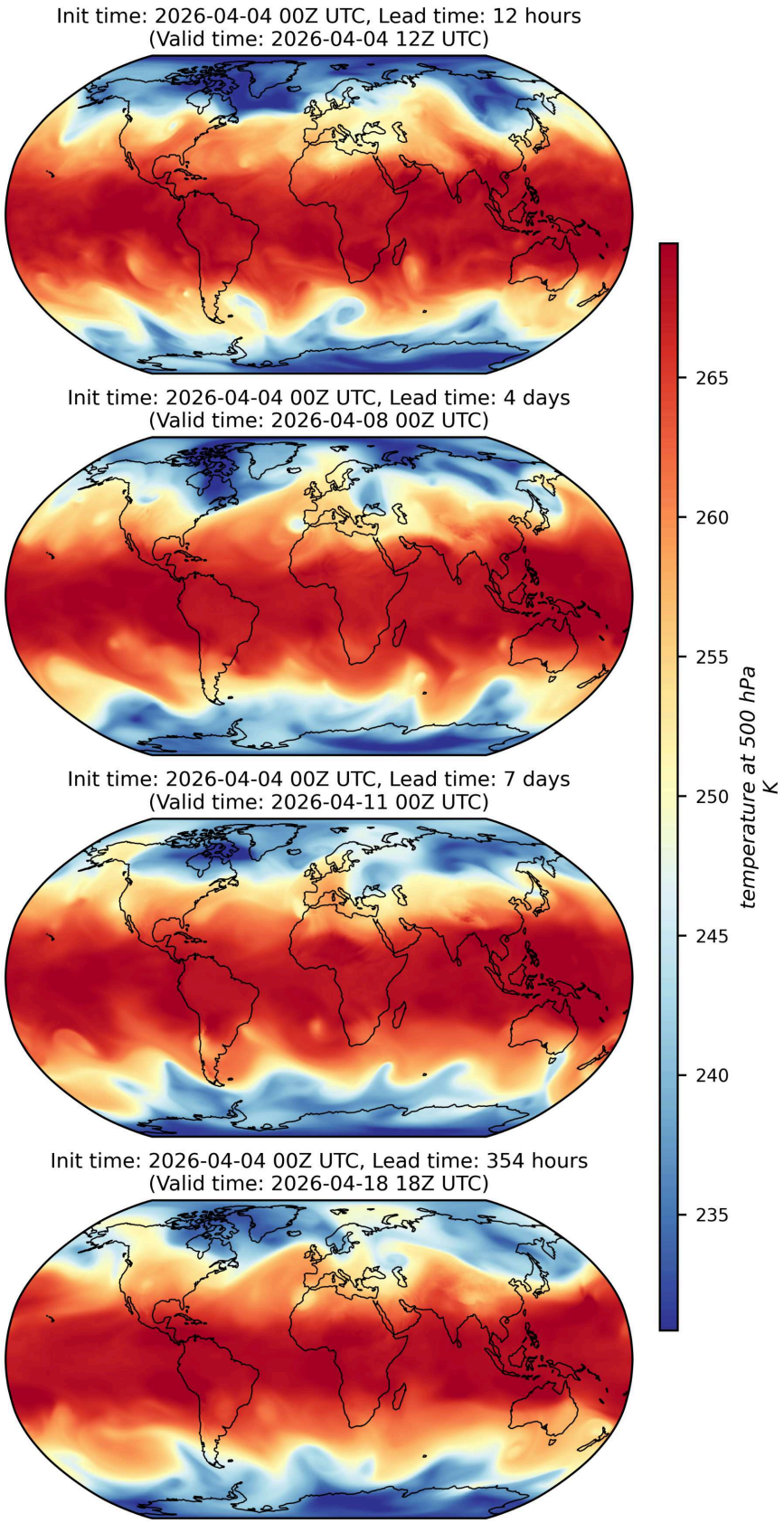


Figure 20: Temperature at 500 hPa GraphMet forecast example.

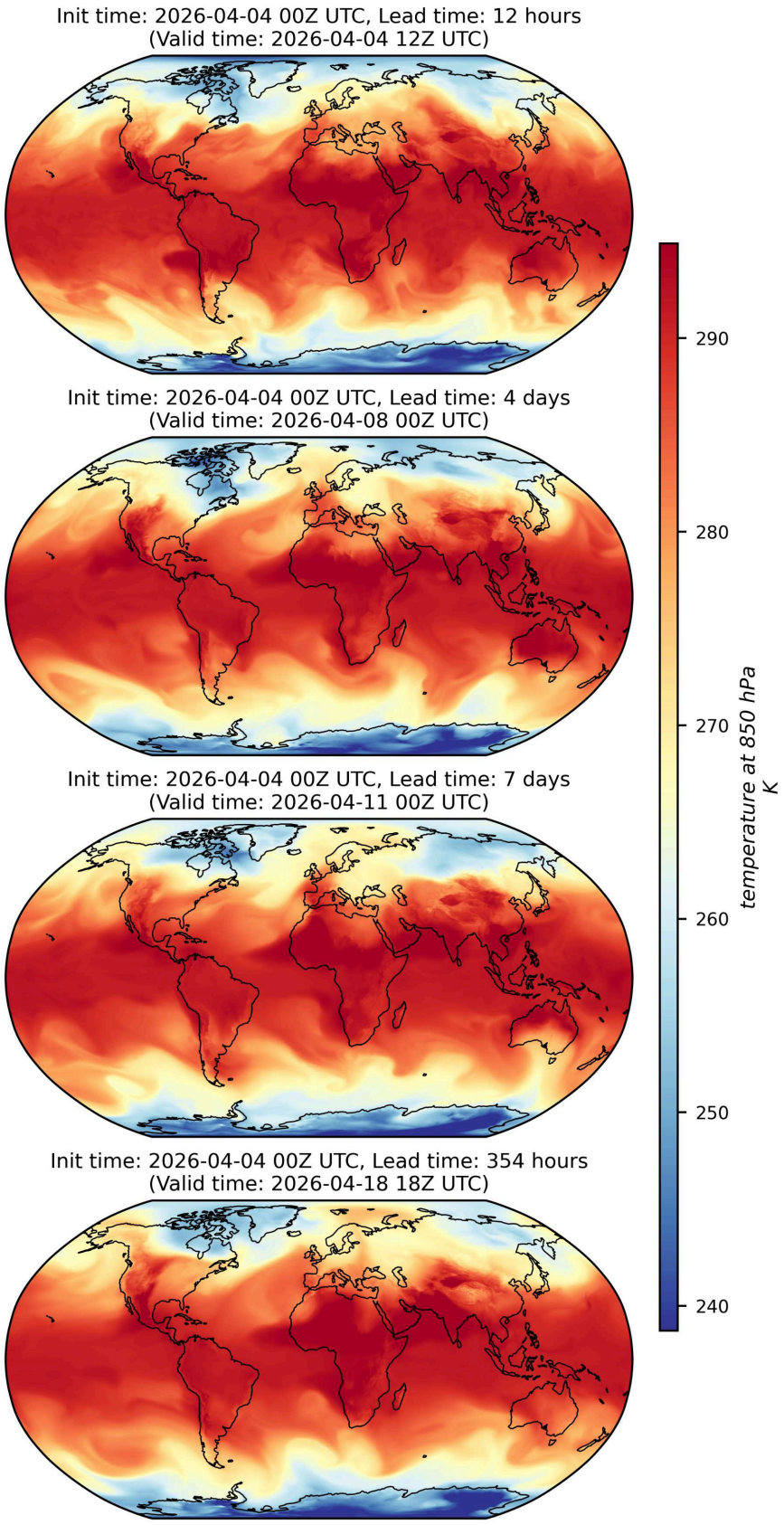


Figure 21: Temperature at 850 hPa GraphMet forecast example.

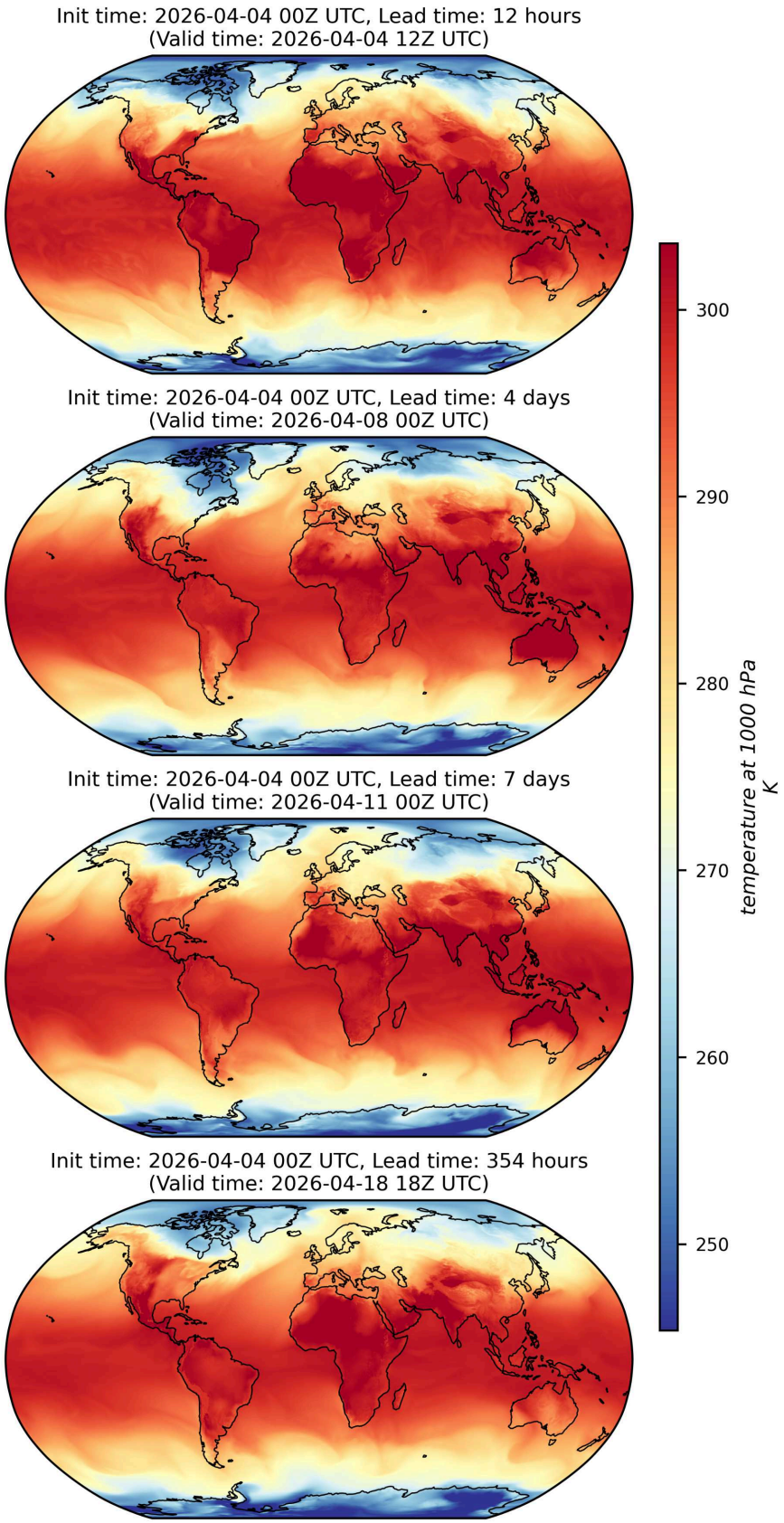


Figure 22: Temperature at 1000 hPa GraphMet forecast example.

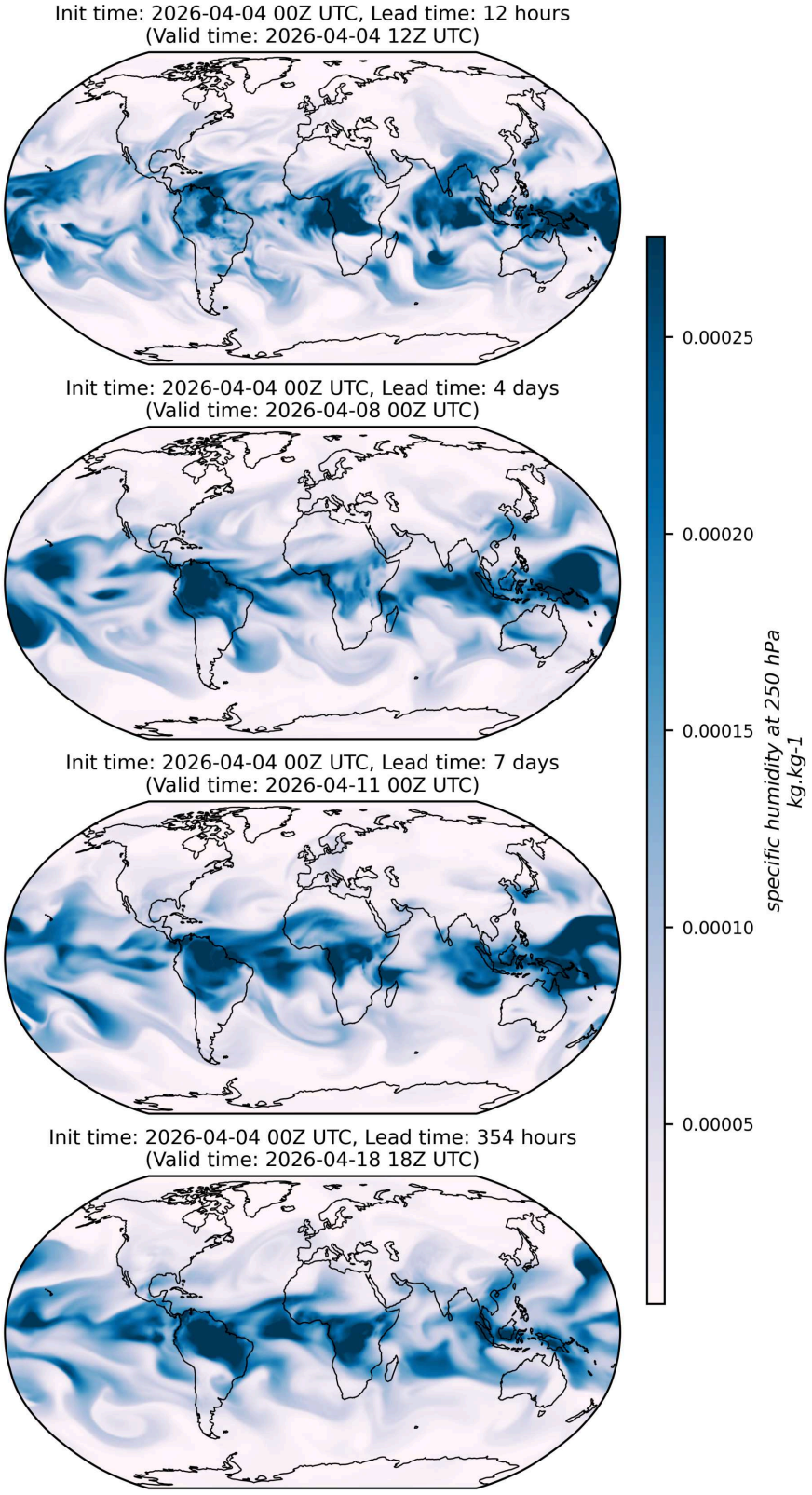


Figure 23: Specific Humidity at 250 hPa GraphMet forecast example.

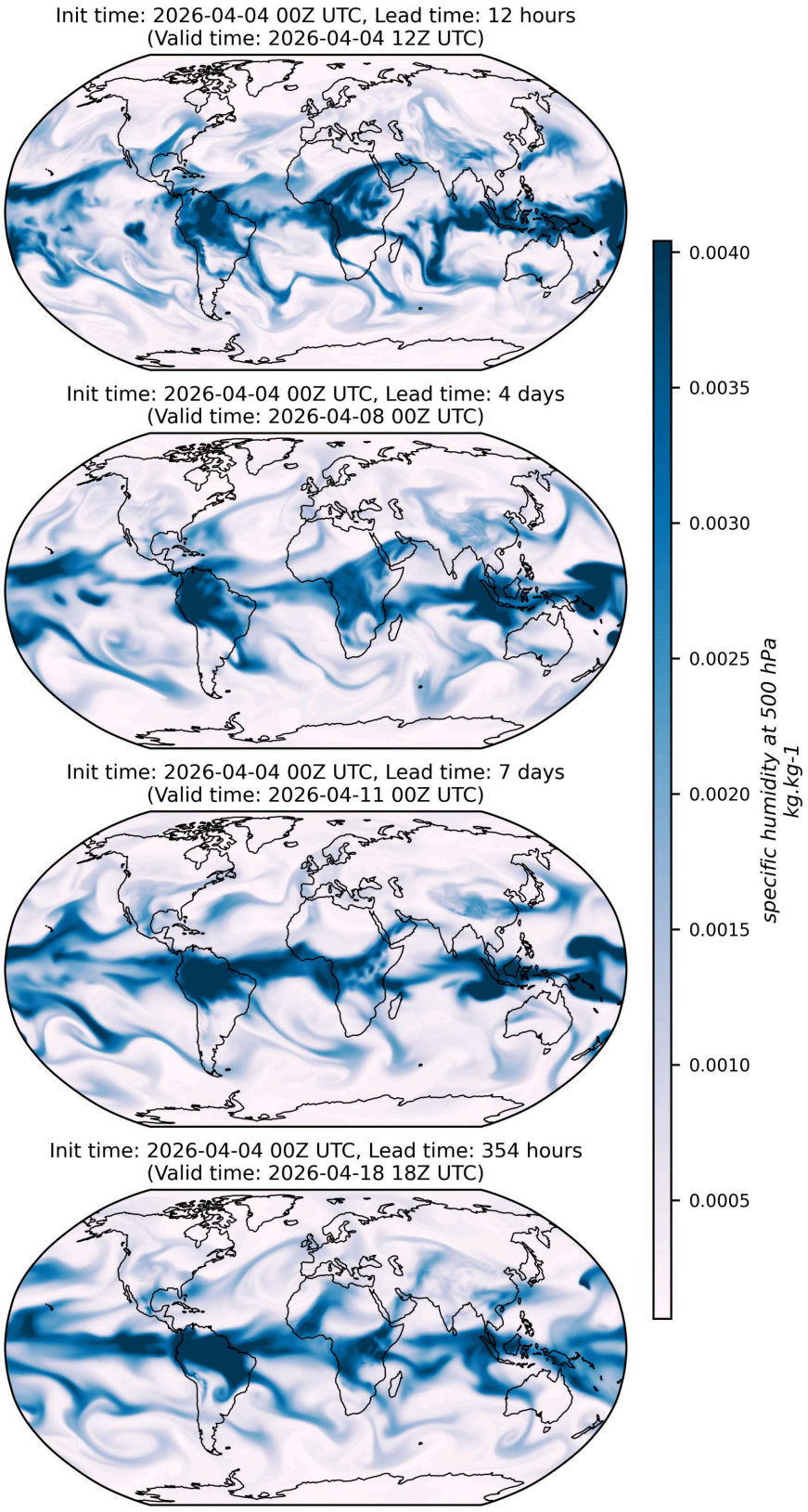


Figure 24: Specific Humidity at 500 hPa GraphMet forecast example.

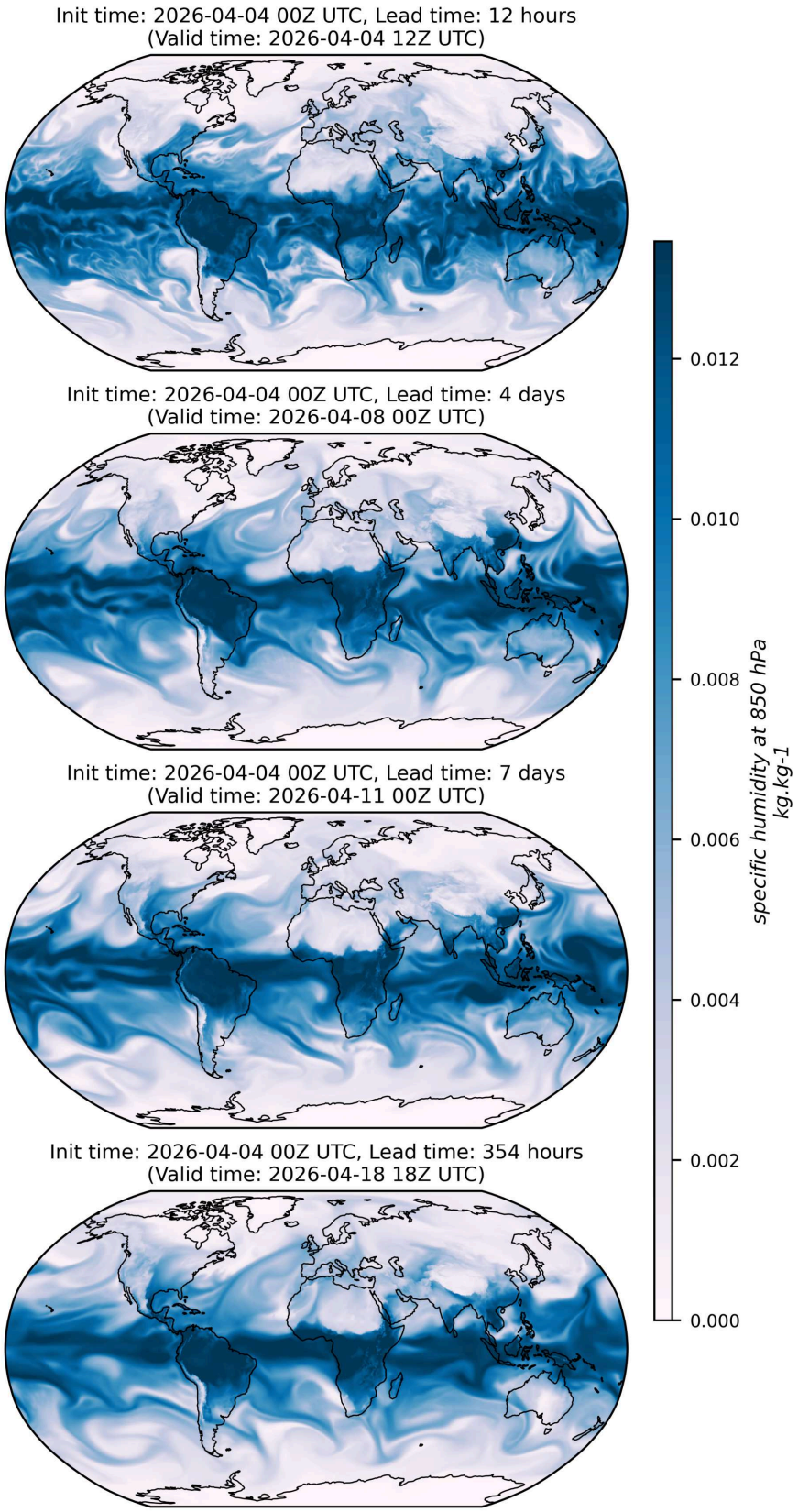


Figure 25: Specific Humidity at 850 hPa GraphMet forecast example.

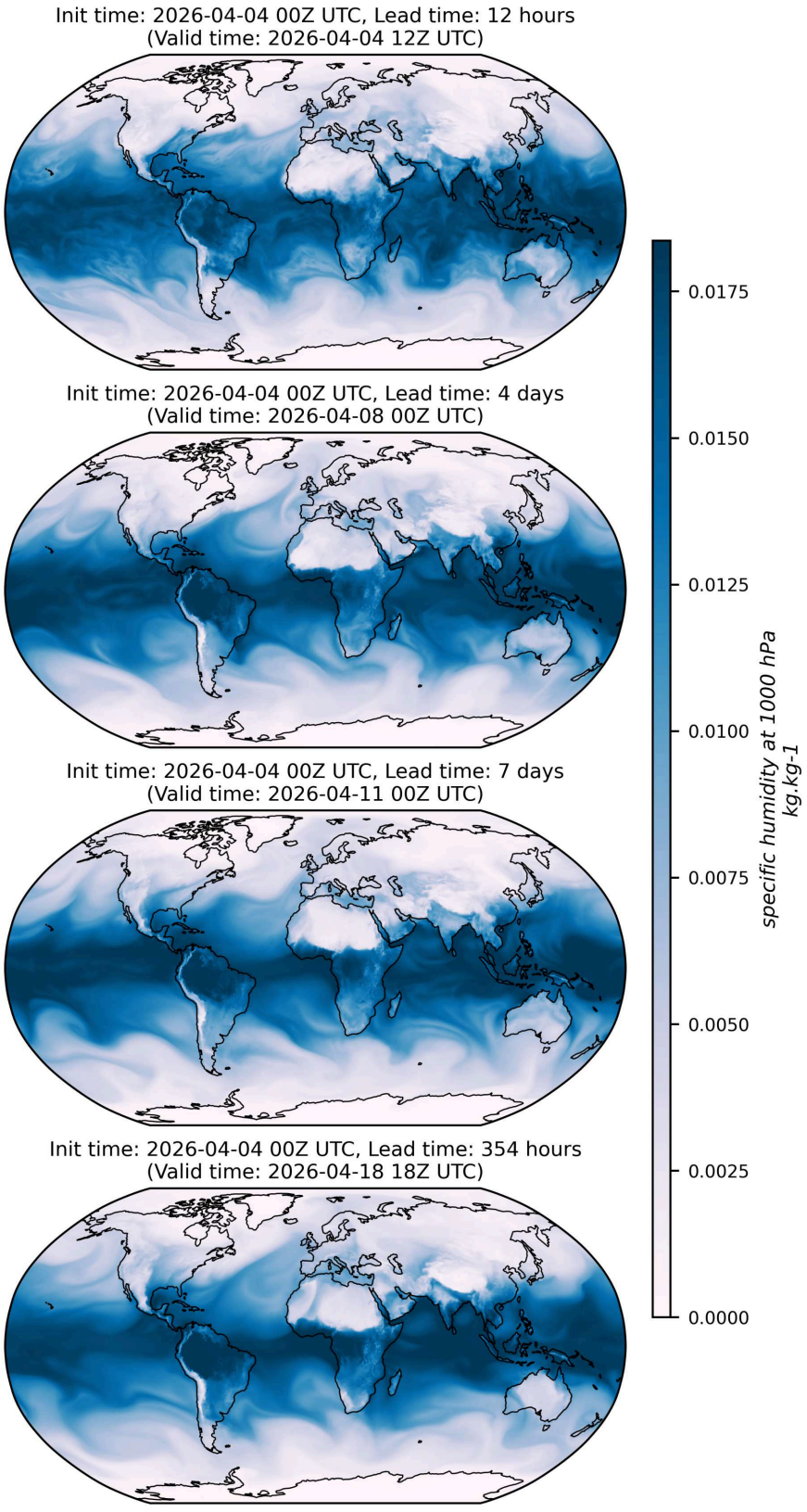


Figure 26: Specific Humidity at 1000 hPa GraphMet forecast example.

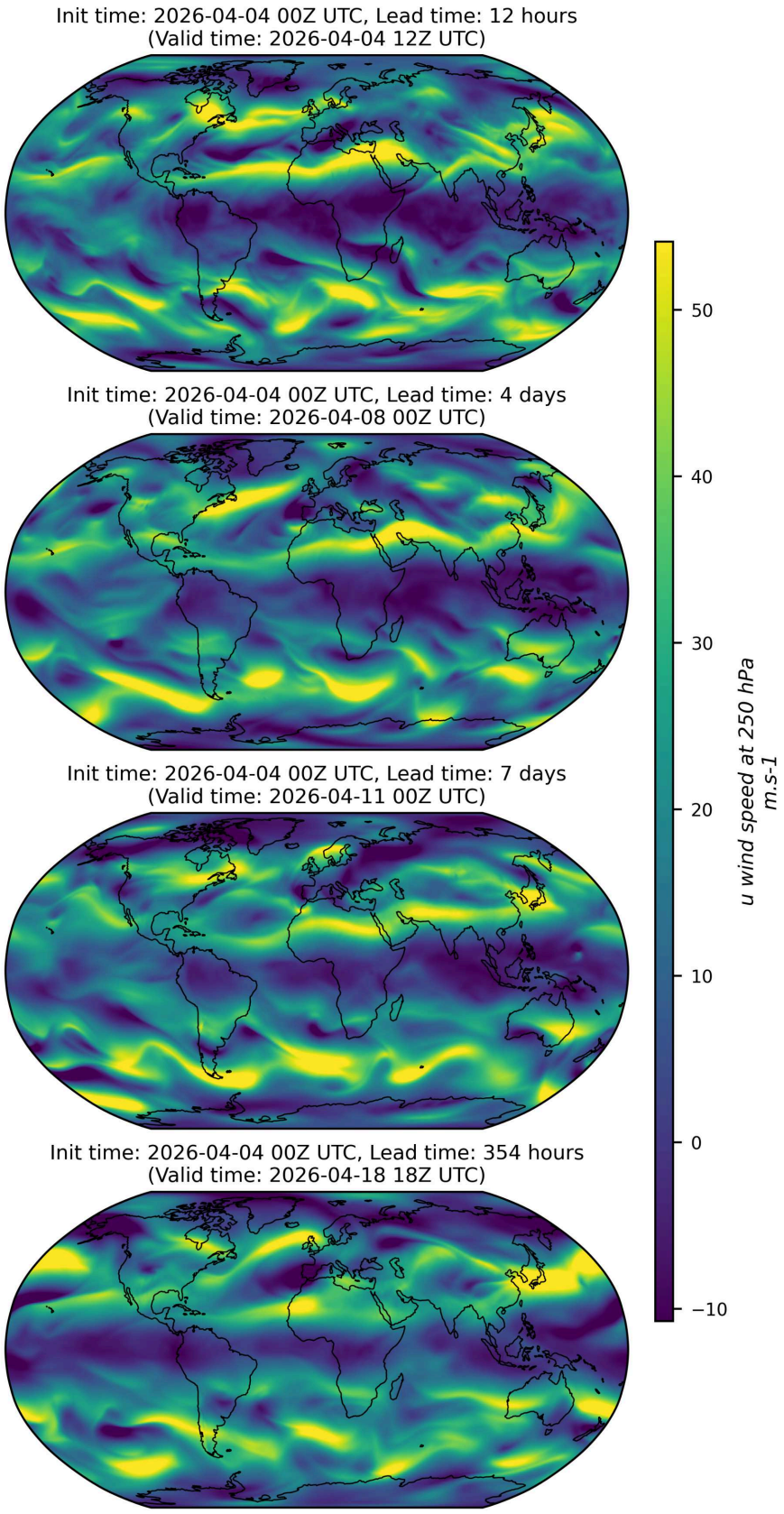


Figure 27: U wind speed at 250 hPa GraphMet forecast example.

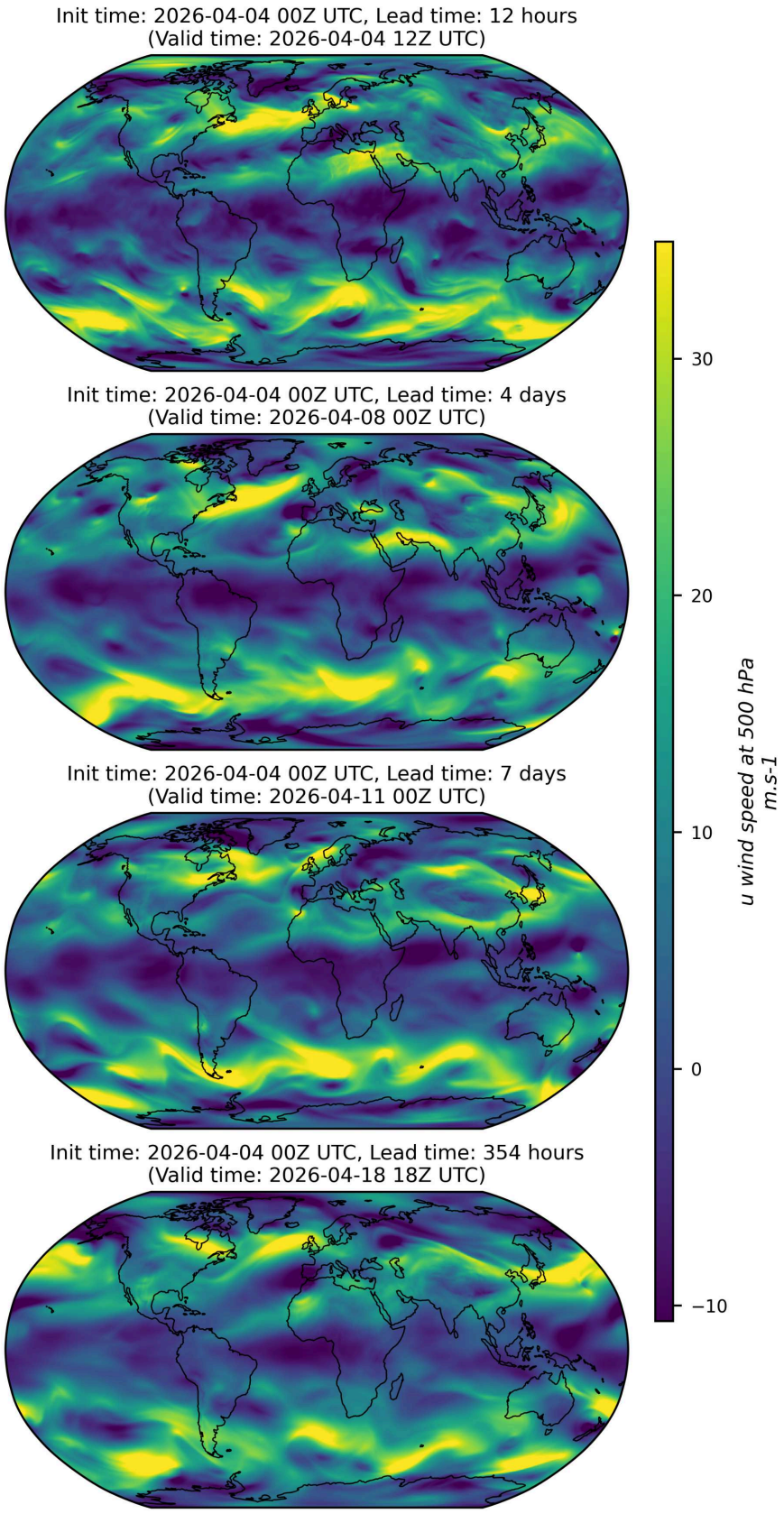


Figure 28: U wind speed at 500 hPa GraphMet forecast example.

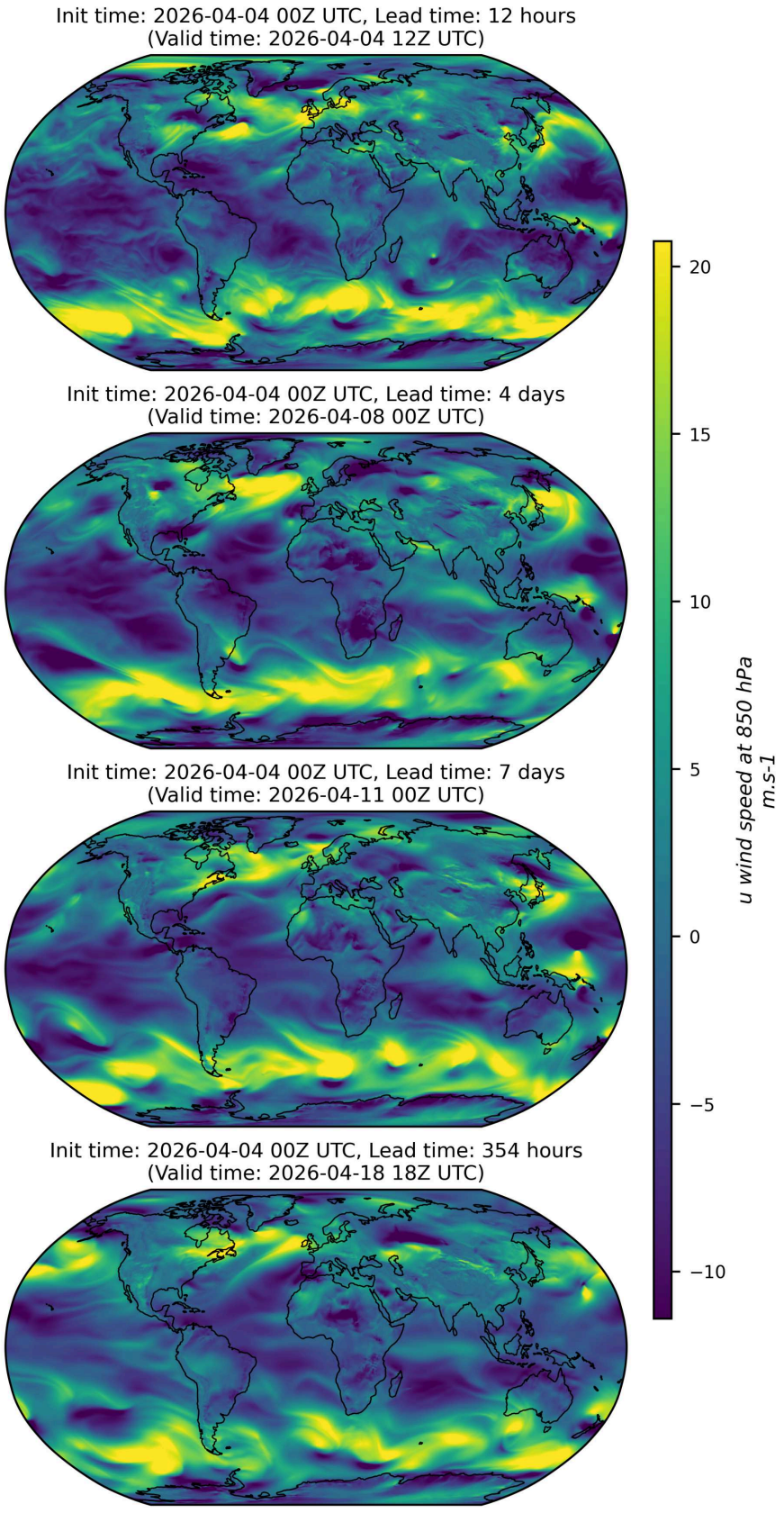


Figure 29: U wind speed at 850 hPa GraphMet forecast example.

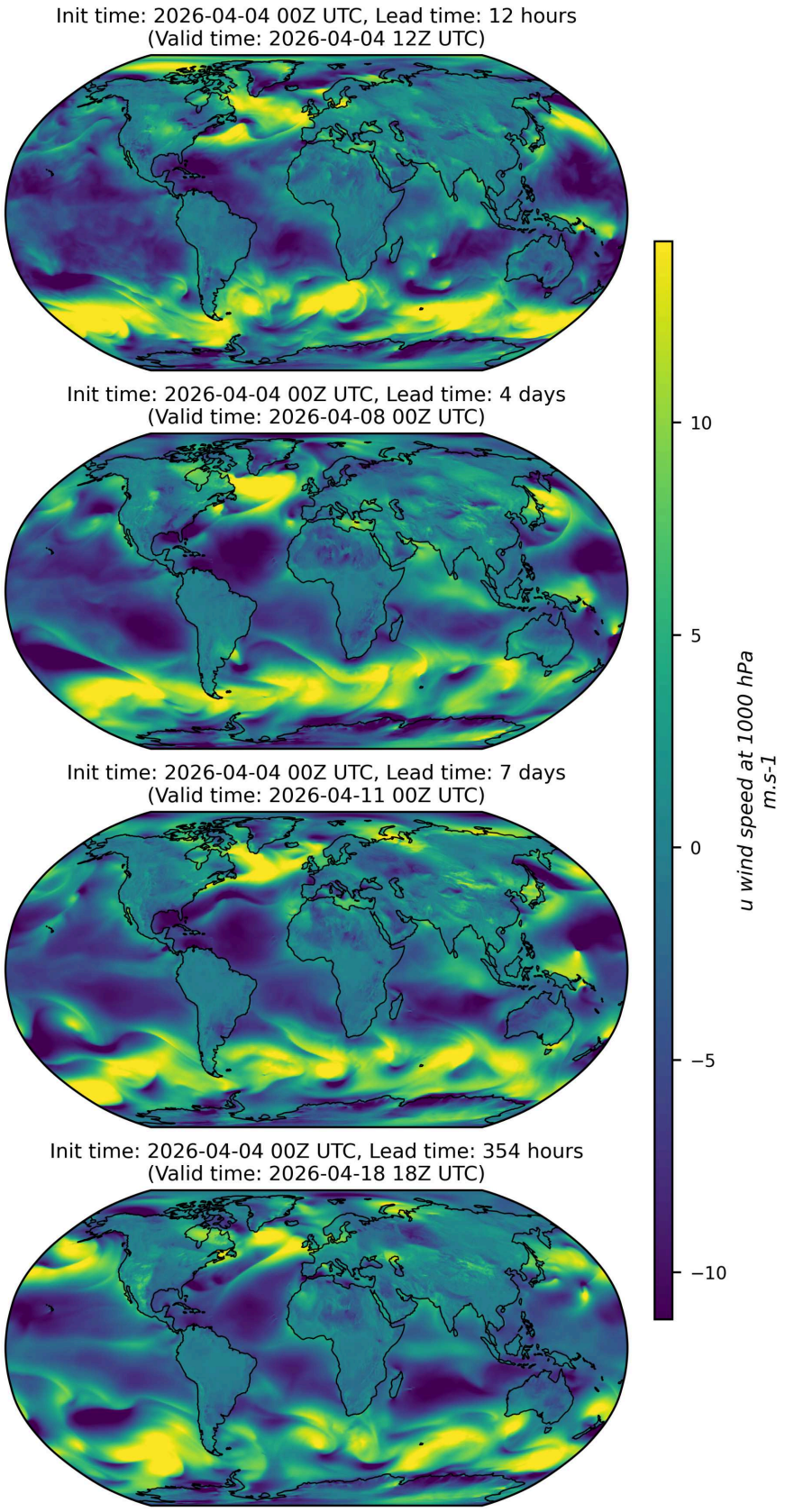


Figure 30: U wind speed at 1000 hPa GraphMet forecast example.

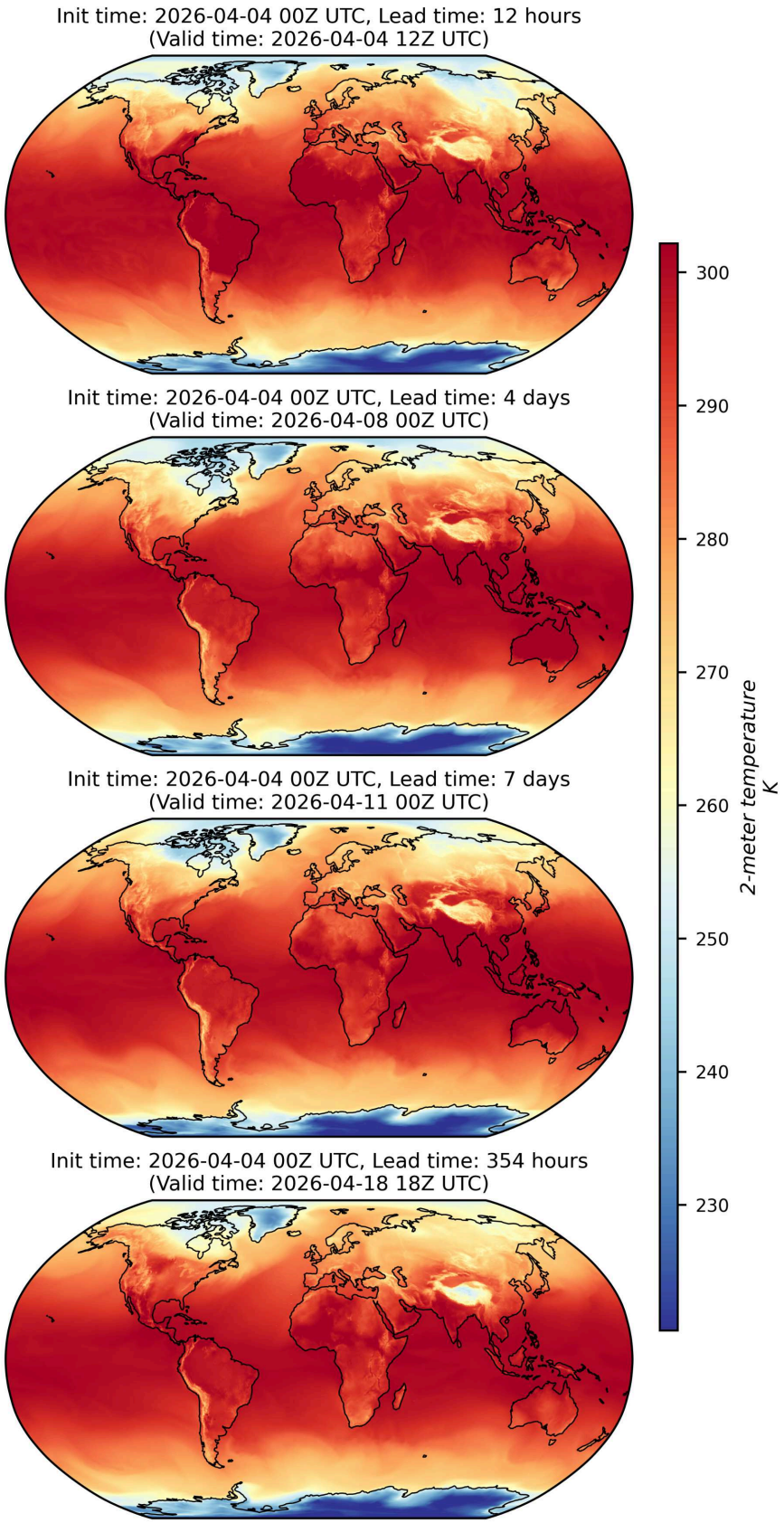


Figure 31: 2-meter temperature GraphMet forecast example.

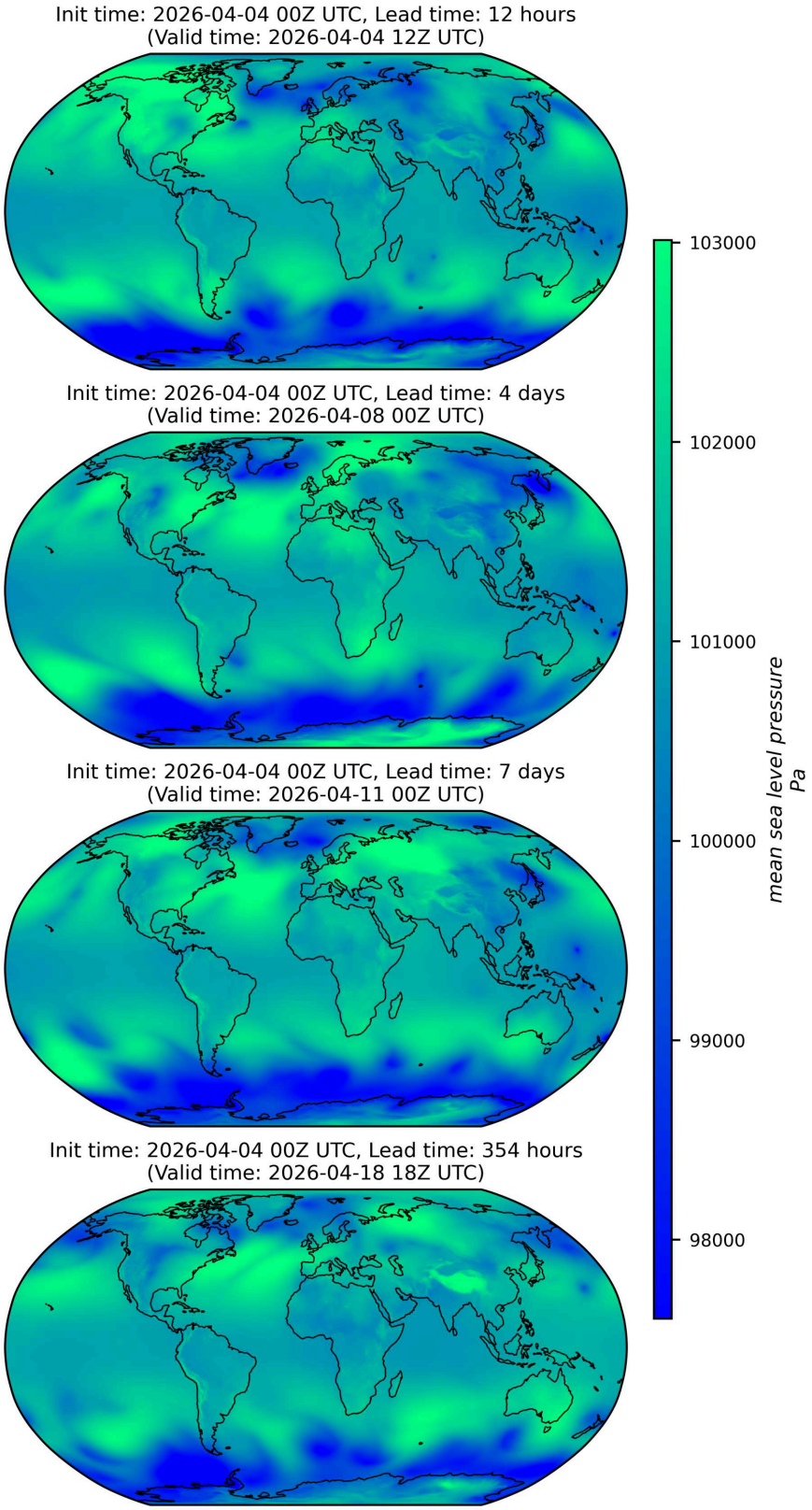


Figure 32: Mean-sea level pressure GraphMet forecast example.

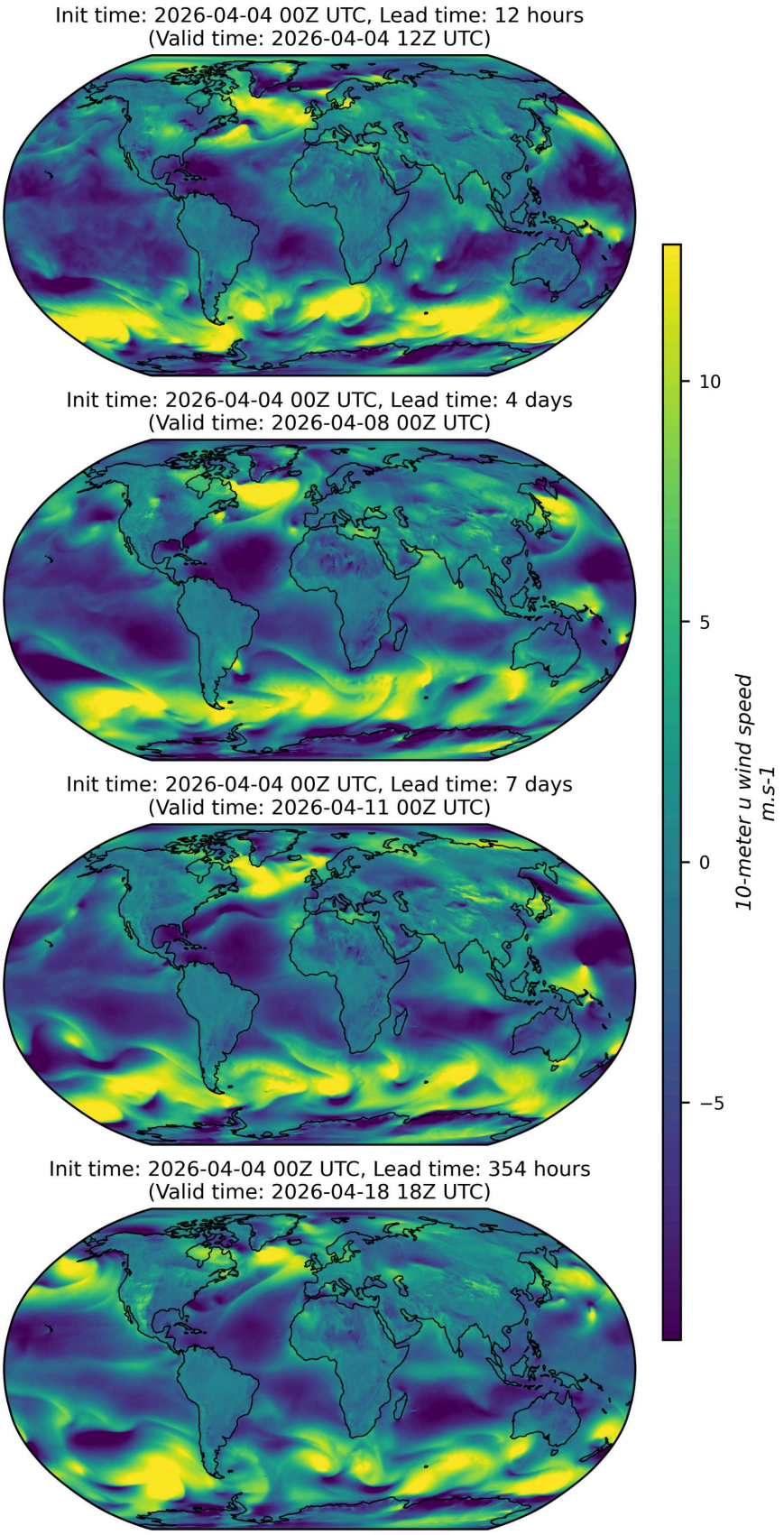


Figure 33: 10-meter u wind speed GraphMet forecast example.

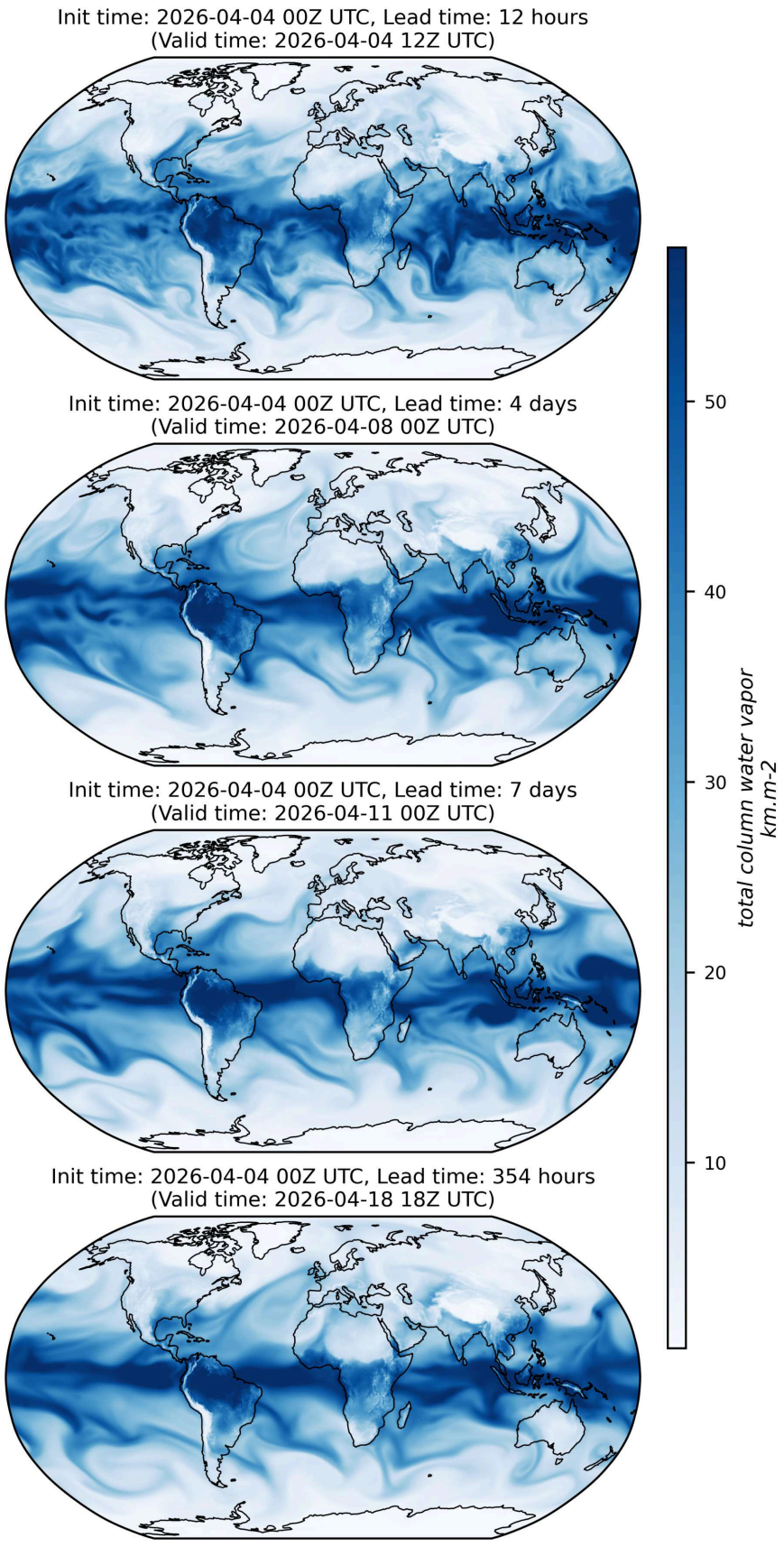


Figure 34: Total column water vapor GraphMet forecast example.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions: replacing GraphCast's nonlinear GNN processor with a single linear Koopman operator, with theoretical guarantees and empirical validation on ERA5/CERRA/MEPS.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed, including sensitivity to batch size for the isotropy penalty, saturation of performance gains beyond $m = 32$ embedding dimension.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All theoretical results (Theorem on over-smoothing, Corollary on latent covariance collapse, Theorem on EYM optimality) are stated with full assumptions and complete proofs provided in Appendices A.1, A.2, and A.3.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The architecture, hyperparameters, training procedure, and dataset specification are fully described in the appendix. Code is available at `https://anonymous.4open.science/w/GraphMet-54ED/`.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Code is released at https://anonymous.4open.science/w/GraphMet-54ED/. ERA5/CERRA data is publicly accessible via the Copernicus Climate Data Store.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Full training hyperparameters (optimizer, learning rate schedule, batch size, hardware) are provided in Appendix C. Data splits (20 years training, 2022 test set) and ERA5 variable specification are detailed in Appendix B.1.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Standard-deviation are reported for all results; also sensitivity tables by varying one hyperparameter at a time provide partial robustness evidence.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute resources are specified in the training hyperparameter table in Appendix C: 64 $\times$ A100 80 GB GPUs for approximately 72 hours for Stage 1, plus 50k additional steps for Stage 2 regional fine-tuning.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research involves weather forecasting using publicly available reanalysis data and conforms with the NeurIPS Code of Ethics; anonymity is preserved via the anonymized code repository.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses positive societal impacts (improved extreme weather forecasting, early detection of tropical cyclones) and acknowledges potential risks of over-reliance on AI forecasting systems without adequate uncertainty quantification.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The model is a weather forecasting system trained on publicly available reanalysis data and poses no misuse risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

Justification: ERA5 is cited and accessed via the Copernicus Climate Data Store under its standard license. GraphCast and all other referenced datasets (CERRA, MEPS) and codebases are cited with their original papers.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The anonymized code repository is released at `https://anonymous.4open.science/w/GraphMet-54ED/` give the full access new tool of procesing and building dataset also with architecture details, training scripts, and hyperparameter documentation provided in the appendix.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects; all experiments use reanalysis datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not used as any important, original, or non-standard component of the core methodology; the method is based on Koopman operator theory and graph neural networks.

    Guidelines:

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