SPHERICAL LAPLACE NEURAL OPERATORS WITH SPATIO-TEMPORAL MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operators provide efficient surrogate models for dynamical partial differential equations (PDEs). Existing spherical neural operators have advanced the modeling of global spatial interactions through spherical spectral representations, while transient temporal dynamics remain implicit in learned discrete-time mappings. To address this limitation, we introduce the Spherical Laplace Neural Operator (SLNO), which combines spherical Fourier modeling in space with Laplace-domain modeling in time. By conditioning temporal transfer functions on the Laplace–Beltrami spectrum, SLNO explicitly represents scale-dependent decay and oscillatory dynamics through learnable poles and channel-valued residues. We further derive analytic initial-state and trajectory-driven responses to construct and propagate internal temporal features within nonlinear neural operator blocks. Extensive experiments on a controlled linear fast–slow system, spherical PDEs and global weather forecasting demonstrate the accurate and stable long-horizon prediction of SLNO. These results demonstrate the potential of explicit spatio-temporal spectral modeling for learning long-term spherical dynamics.

1 INTRODUCTION

Neural operators learn solution maps of partial differential equations (PDEs), providing an alternative to repeated numerical simulation (Chen & Chen, 1995; Lu et al., 2019; 2021; Li et al., 2020a; Kovachki et al., 2023; Raissi et al., 2019). For dynamical systems on the sphere, their practical value depends not only on next-step accuracy but also on reliability when predictions are repeatedly reused as inputs (Brandstetter et al., 2022). The forecasting operator aims to capture spherical spatial interactions while retaining predictive accuracy over long autoregressive rollouts.

Fourier neural operators (FNOs) parameterize global interactions in the Fourier domain (Li et al., 2020a). Later spectral architectures replace or reorganize that transform with wavelets, dimension-wise factorization, U-shaped multiscale skips, or a geometry-informed latent grid (Gupta et al., 2021; Tripura & Chakraborty, 2023; Tran et al., 2021; Rahman et al., 2022; Li et al., 2023), while graph kernel networks discretize the same integral-operator framework (Li et al., 2020b; Kovachki et al., 2023). Spherical Fourier neural operators (SFNOs) introduce spherical harmonics and degree-dependent spectral transformations (Bonev et al., 2023), while generalized spherical neural operators (GSNOs) extend spherical interactions through a Green-function formulation (Tang et al., 2026). Adaptive Fourier layers have been carried into data-driven global weather models (Guibas et al., 2021; Pathak et al., 2022), and graph-based forecasters trained on reanalysis now target medium-range prediction (Lam et al., 2023; Hersbach et al., 2020). Laplace neural operators (LNOs) offer a complementary temporal representation based on learned poles and residues (Cao et al., 2024).

Despite these advances, short-horizon accuracy and long-horizon reliability can diverge (Brandstetter et al., 2022). A predictor with low one-step error can deteriorate when repeatedly evaluated on its own outputs: approximation errors alter subsequent inputs, and the learned map can amplify the resulting perturbations and neglect transient temporal dynamics. Accurate predictions on isolated training pairs therefore do not fully characterize autoregressive behavior (Brandstetter et al., 2022; Cao et al., 2024). The challenge is to learn spherical forecasting operators that maintain low prediction error under repeated composition, rather than merely improve the next-step fit. We investigate explicit temporal structure as an inductive bias for this problem. Different spherical scales can sup-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

port different temporal responses, and slowly varying and rapidly oscillating components can coexist within the same scale.

We introduce the **Spherical Laplace Neural Operator (SLNO)** for long-horizon spherical forecasting. SLNO parameterizes the global interaction through scale-conditioned temporal response families, rather than only degree-wise spectral transformations. A shared spectral network maps Laplace–Beltrami eigenvalues to temporal poles and channel-response matrices, representing multiple latent dynamical branches at each degree and linking their parameters across degrees. An initial-response block constructs an internal trajectory from a single encoded state. Subsequent trajectory-driven blocks apply analytic responses to its finite temporal Fourier representation, interleaved with nonlinear channel mixing and residual connections. This construction makes temporal response structure explicit within each prediction step while retaining the standard outer autoregressive interface and requiring no intermediate-time labels. SLNO therefore investigates whether making the within-step response family explicit provides a useful inductive bias for repeated prediction. Our contributions include:

- **Scale-conditioned temporal representation.** We formulate spherical global interactions through temporal transfer functions whose poles and channel responses are generated by a shared spectral law, organizing latent dynamics within and across spatial scales.
- **Trajectory-based operator architecture.** We develop initial-state and trajectory-driven blocks that process internally generated temporal features through analytic responses and nonlinear mixing within a single-state forecasting interface.
- **Long-horizon stability.** We separate one-step accuracy from long-rollout behavior on controlled fast–slow dynamics, spherical shallow water, and a matched partial-autoregressive global weather forecast.

2 Related Work

Neural operators and spherical spectral modeling. Neural operators learn mappings between function spaces and have emerged as efficient surrogate models for parametric PDEs. Representative approaches include DeepONet (Lu et al., 2019; 2021), Fourier Neural Operators (FNOs) (Li et al., 2020a), graph-based neural operators (Li et al., 2020b;c), and subsequent spectral architectures based on wavelets, multiscale representations, and geometric transformations (Gupta et al., 2021; Tripura & Chakraborty, 2023; Rahman et al., 2022; Li et al., 2023). For spherical domains, early geometric learning approaches introduced rotation-equivariant spherical convolutions (Cohen et al., 2018), graph-based spherical representations (Defferrard et al., 2020), and gauge-equivariant convolutional networks (Cohen et al., 2019). Building upon spherical harmonic analysis, Spherical Fourier Neural Operators (SFNOs) (Bonev et al., 2023) extend Fourier operator learning to spherical geometries through degree-dependent spectral transformations, demonstrating effective global modeling and stable autoregressive simulation of atmospheric dynamics. Generalized Spherical Neural Operators (GSNOs) (Tang et al., 2026) further develop a Green's-function-based formulation to enhance the flexibility of spherical spatial interactions, including the modeling of non-equivariant physical systems.

Temporal operator learning and long-term prediction. Temporal dynamics have been modeled through continuous-time operator representations, structured state-space models, and learned autoregressive evolution. Laplace Neural Operators (LNOs) (Cao et al., 2024) introduce a pole–residue formulation for learning temporal responses, while Geometric Laplace Neural Operators (GLNOs) (Tang et al., 2025) extend Laplace-based operator learning to geometric domains. Structured state-space models, including S4 (Gu et al., 2021), provide efficient representations of long-range temporal dependencies. More recent developments, such as State Space Neural Operators (SS-NOs) (Koren & Lanthaler, 2025) and Mamba Neural Operators (Cheng et al., 2025), further investigate the integration of state-space modeling with PDE solution operators. In parallel, autoregressive neural PDE solvers have addressed long-horizon error accumulation through training and architectural strategies (Brandstetter et al., 2022), while data-driven weather forecasting architectures, including FourCastNet (Pathak et al., 2022), GraphCast (Lam et al., 2023), and Pangu-Weather (Bi et al., 2023), demonstrate the practical importance of accurate temporal evolution in large-scale physical systems.

3 PRELIMINARIES

3.1 SPHERICAL HARMONICS

The complex spherical harmonics Y_ℓ^m, indexed by degree $\ell \geq 0$ and order $-\ell \leq m \leq \ell$, form an orthonormal basis of $L^2(\mathbb{S}^2)$ (Driscoll & Healy, 1994). For a square-integrable field f, the spherical harmonic transform (SHT) and its inverse are

$$\widehat{f}_{\ell m} = \int_{\mathbb{S}^2} f(\omega)\overline{Y_\ell^m(\omega)}\, d\omega, \tag{1}$$

$$f(\omega) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \widehat{f}_{\ell m} Y_\ell^m(\omega), \tag{2}$$

where $d\omega$ is the surface-area measure on the unit sphere. The inverse expansion converges in L^2; numerical representations retain a finite band $0 \leq \ell \leq L$. For channel-valued fields, both transforms act componentwise. Spherical harmonics satisfy

$$-\Delta_{\mathbb{S}^2} Y_\ell^m = \lambda_\ell Y_\ell^m, \qquad \lambda_\ell = \ell(\ell+1). \tag{3}$$

The Laplace–Beltrami eigenvalue depends only on degree and provides a spectral coordinate for spherical spatial scale. For a matrix-valued zonal kernel depending on $\omega \cdot \eta$, the spherical convolution theorem gives a degree-wise spectral action (Bonev et al., 2023):

$$\begin{aligned} z(\omega) &= \int_{\mathbb{S}^2} \kappa(\omega \cdot \eta) v(\eta)\, d\eta, \\ \widehat{z}_{\ell m} &= W_\ell \widehat{v}_{\ell m}. \end{aligned} \tag{4}$$

Here $W_\ell \in \mathbb{C}^{C_{\text{out}} \times C_{\text{in}}}$ includes the kernel's spectral multiplier and transform normalization. It mixes channels while acting identically on every order m at fixed degree ℓ. This structure characterizes the isotropic linear convolution branch.

3.2 LAPLACE TRANSFORM AND TEMPORAL CONVOLUTION

For a locally integrable causal signal $x(t)$ of exponential order, the unilateral Laplace transform is

$$X(s) = \mathcal{L}[x](s) = \int_0^\infty x(t) e^{-st}\, dt, \qquad \operatorname{Re} s > \sigma_0, \tag{5}$$

where $s \in \mathbb{C}$ and σ_0 specifies a convergence half-plane. For a causal matrix-valued kernel κ and a vector-valued input x, the zero-state response and its Laplace representation are

$$z(t) = (\kappa * x)(t) = \int_0^t \kappa(t-\sigma) x(\sigma)\, d\sigma, \tag{6}$$

$$Z(s) = K(s)X(s), \qquad K(s) = \mathcal{L}[\kappa](s). \tag{7}$$

The convolution identity holds on a common right half-plane of absolute convergence. The matrix-valued transfer function describes the causal linear input–output response used in Laplace neural operators (Cao et al., 2024). Together, SHT and the Laplace transform convert the structured spatial and temporal convolutions into spectral-domain actions while retaining channel coupling.

4 METHODOLOGY

We study autonomous spherical dynamics with fixed system parameters a,

$$\partial_\tau u(\omega, \tau) = \mathcal{F}_a[u(\cdot, \tau)](\omega), \qquad u(\omega, 0) = u_0(\omega). \tag{8}$$

The forecasting task is to approximate $\mathcal{G}_\Delta^\dagger : u_0 \mapsto u(\cdot, \Delta)$ and repeatedly apply the learned map over longer horizons. With fixed training-data normalization $\mathcal{A}$, we encode

$$x_0 = \mathcal{A}(u_0), \qquad h = E_\theta(x_0), \qquad h : \mathbb{S}^2 \to \mathbb{R}^C. \tag{9}$$

The internal coordinate $t \in [0, T]$ is a latent axis within one forecast call. Its endpoint is decoded as the physical forecast at Δ; intermediate coordinates are not assumed to be physical solution states.

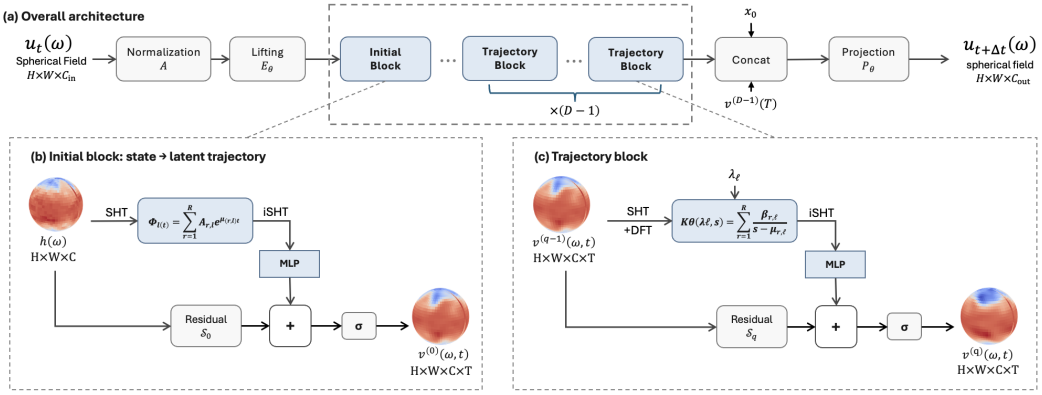

Figure 1: SLNO architecture. (a) The forecasting step maps the input spherical field through normalization and lifting, one initial block, and $D-1$ trajectory blocks, then concatenates the input state and projects back to the next spherical field. (b) The initial block builds a latent trajectory from a single encoded state. (c) Each trajectory block applies a scale-dependent response in the joint spherical-harmonic and temporal-Fourier domain. A nonlinear post-response transformation, consisting of normalization and pointwise channel mixing, acts on that response, and its output is added to a residual connection.

4.1 Spherical–Laplace Operator

We parameterize the trajectory-driven linear global branch by a causal spherical–temporal convolution,

$$(\mathcal{K}_\theta v)(\omega,t)=\int_0^t\int_{\mathbb{S}^2}\kappa_\theta(\omega\cdot\eta,t-\sigma)v(\eta,\sigma)\,d\eta\,d\sigma. \tag{10}$$

The zonal spatial kernel defines the isotropic linear global branch, which is subsequently embedded within the full nonlinear SLNO architecture. Spherical reduction and the Laplace convolution theorem give

$$Z_{\ell m}(s)=K_{\theta,\ell}(s)V_{\ell m}(s),\qquad K_{\theta,\ell}(s)=\mathcal{L}[\kappa_{\theta,\ell}](s), \tag{11}$$

on a common convergence half-plane. The channel-valued transfer is shared over orders m at fixed degree ℓ; the reduction is detailed in Appendix A.

Following the pole–residue representation of temporal operator learning (Cao et al., 2024), we define

$$K_{\theta,\ell}(s)=K_\theta(\lambda_\ell,s)=\sum_{r=1}^{R}\frac{B_{\theta,r}(\lambda_\ell)}{s-\mu_{\theta,r}(\lambda_\ell)},\qquad \lambda_\ell=\ell(\ell+1). \tag{12}$$

Here $\mu_{r,\ell}\in\mathbb{C}$ specifies a latent temporal branch and $B_{r,\ell}\in\mathbb{C}^{C\times C}$ its channel response. This strictly proper rational family contains at most R simple poles. Pole real and imaginary parts control exponential growth/decay and angular oscillation frequency. Multiple poles represent coexisting temporal branches at a degree. Within each block, one dictionary determines the response at every retained time, rather than assigning independent transformations to individual time nodes.

Appendix records the shared spectral parameterization of each block dictionary.

4.2 SLNO Block

SLNO first constructs a latent trajectory from the encoded state, then processes it through successive responses and nonlinear mixing. We denote linear global responses by $z^{(q)}$ and complete nonlinear block outputs by $v^{(q)}$.

Initial block. For $\widehat{h}=\mathrm{SHT}[h]$, define

$$\widehat{z}^{(0)}_{\ell m}(t)=\Phi^{(0)}_\ell(t)\widehat{h}_{\ell m},\qquad \Phi^{(0)}_\ell(t)=\sum_{r=1}^{R}A^{(0)}_{r,\ell}e^{\mu^{(0)}_{r,\ell}t}. \tag{13}$$

We evaluate this family at $t_j = j\delta t$, $j = 0, \ldots, N_t - 1$, with $T = (N_t - 1)\delta t$, and apply inverse SHT to obtain a time-indexed latent field. The intermediate states are generated from the input alone and receive no direct physical supervision. The initial response is a latent input–output map; its normalization and relation to physical propagation are discussed in Appendix B.

Trajectory block. For block $q \geq 1$, SHT is applied to each time slice of $v^{(q-1)}$. Its forward-normalized temporal Fourier coefficients are

$$\alpha^{(q)}_{k,\ell m} = \frac{1}{N_t} \sum_{j=0}^{N_t-1} \widehat{v}^{(q-1)}_{\ell m}(t_j) e^{-2\pi \mathrm{i} kj/N_t}, \qquad k = 0, \ldots, N_t - 1. \tag{14}$$

Here $\alpha^{(q)}_{k,\ell m} \in \mathbb{C}^C$. The frequencies use signed DFT ordering, $\omega_k = 2\pi k/(N_t \delta t)$. For a learned pole μ and excitation ξ, let

$$\psi_t(\mu, \xi) = \frac{e^{\mu t} - e^{\xi t}}{\mu - \xi}, \qquad \psi_t(\mu, \mu) = te^{\mu t}. \tag{15}$$

The trajectory-driven response is

$$\widehat{z}^{(q)}_{\ell m}(t) = \sum_{r=1}^{R} \sum_{k=0}^{N_t-1} B^{(q)}_{r,\ell} \alpha^{(q)}_{k,\ell m} \psi_t\Big(\mu^{(q)}_{r,\ell}, \mathrm{i}\omega_k\Big). \tag{16}$$

The matrix–vector products perform channel mixing. This is the analytic convolution response to the finite trigonometric interpolant of the input trajectory, followed by inverse SHT. The entire retained trajectory supplies the temporal coefficients, so the finite implementation is a whole-window latent calculation, while causality characterizes the underlying continuous kernel. The derivation, collision-safe evaluation, and real-output conventions are given in Appendix B.

4.3 SLNO Architecture

The global responses are embedded in residual blocks:

$$v^{(0)} = \mathcal{S}_0 h + \mathcal{N}_0[z^{(0)}], \tag{17}$$

$$v^{(q)} = \mathcal{S}_q v^{(q-1)} + \mathcal{N}_q[z^{(q)}], \qquad q = 1, \ldots, D - 1. \tag{18}$$

The maps N_q denote nonlinear post-response transformations consisting of normalization and pointwise channel MLPs with GELU activations (Hendrycks & Gimpel, 2016); S_q provides residual grid matching, with an additional time broadcast in the initial block.matching, with an additional time broadcast in the initial block. This broadcast is distinct from the dynamically generated response. Nonlinear mixing is applied at each retained time before the next trajectory block, allowing subsequent responses to depend on transformed within-step features rather than only an endpoint. It can also couple spherical modes between the linear global branches.

One initial block is followed by $D - 1$ trajectory blocks. Intermediate blocks preserve the complete time axis, whereas the last block returns the endpoint at T. In the deterministic, time-separable post-response host used here, endpoint-only evaluation avoids constructing unused output slices without changing the endpoint computation. The decoder combines the final latent state with the normalized input through the host's channel-concatenation skip:

$$F_\theta(x_0) = P_\theta\Big(\mathrm{concat}_C[v^{(D-1)}(T), x_0]\Big). \tag{19}$$

Single-step training and long-horizon rollout. The forecaster is trained on one-step pairs, without autoregressive fine-tuning or intermediate-time labels. Long-horizon predictions use

$$\widehat{x}^{(0)} = x_0, \qquad \widehat{x}^{(n+1)} = F_\theta(\widehat{x}^{(n)}), \qquad \widehat{u}(n\Delta) = \mathcal{A}^{-1}(\widehat{x}^{(n)}). \tag{20}$$

Each call constructs a new internal trajectory for one forecast interval. Temporal structure is introduced within the learned step map, not through a separate long-horizon solver or persistent hidden state. Long-horizon reliability is evaluated directly through autoregressive rollout of the composed nonlinear forecaster.

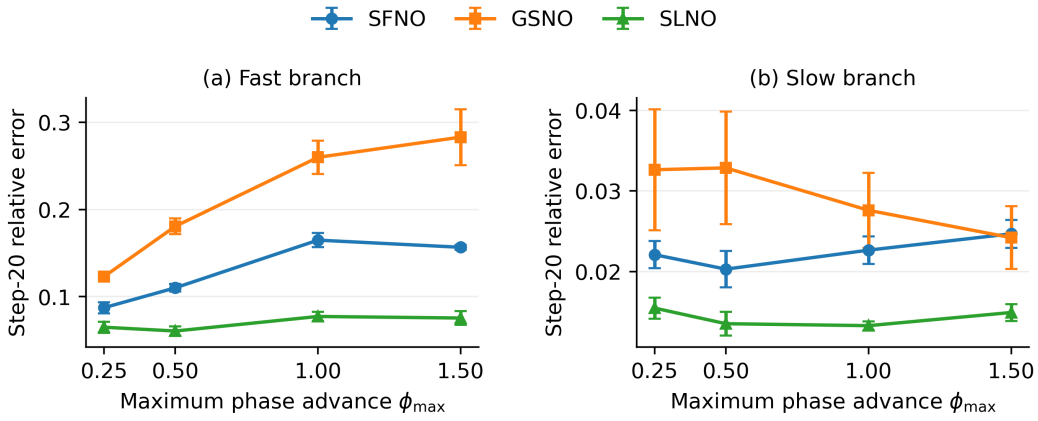

Figure 2: Step-20 relative error as the fast-branch phase advance increases. (a) Fast branch. (b) Slow branch. Points are means over three matched seeds; error bars are sample standard deviations. Lines connect evaluated difficulty levels only. The spatial grid, retained degrees, decay rates, and time step are fixed.

5 Experiments

We evaluate SLNO through three experiments of increasing complexity, progressing from controlled linear dynamics to nonlinear spherical PDEs and real-world atmospheric forecasting. These experiments progressively investigate the effectiveness of explicit spatio-temporal modeling across increasingly challenging spherical evolution problems.

5.1 Controlled Fast–Slow Spherical Dynamics

Motivation and controlled system. Autoregressive rollout provides a direct stress test for learned PDE dynamics because short-horizon accuracy alone does not characterize behavior under repeated composition (Brandstetter et al., 2022; Bonev et al., 2023). To isolate temporal difficulty from spatial resolution and slow-mode dynamics, we construct a controlled linear spherical system with one slow mode and a fast conjugate pair at every retained degree. Reference trajectories are generated by exact modal propagation of the known dynamics. Appendix E records the modal equations and data-generation procedure.

Difficulty and protocol. The fast frequency and per-step phase are $\omega_\ell = c\sqrt{\lambda_\ell}$ and $\phi_\ell = \omega_\ell \Delta t$. We use the maximum per-step phase advance $\phi_{\max} = c\sqrt{\lambda_{\ell_{\max}}}\Delta t$ as the difficulty coordinate and evaluate $\phi_{\max} \in \{0.25, 0.50, 1.00, 1.50\}$. Only c changes across the sweep. The 32×64 equiangular grid, degrees $1 \le \ell \le 15$, $\gamma_s = 0.05$, $\gamma_f = 0.10$, $\Delta t = 0.06$, and the slow temporal law remain fixed. The sweep therefore increases the temporal phase advance of the fast branch without changing the spatial resolution or slow decay. Step 20 is used as the primary long-horizon comparison point and does not correspond to 20 physical hours. Branch norms, checkpoint selection, and the remaining protocol are given in Appendix E.

Fast-branch rollout and slow-branch control. Figure 2 separates the fast and slow branches. From the easiest to the hardest setting, the step-20 fast-branch error increases by factors of 1.793 for SFNO, 2.300 for GSNO, and approximately 1.162 for SLNO. At $\phi_{\max} = 1.50$, SLNO's fast-branch error is 51.8% lower than SFNO and 73.3% lower than GSNO. Although SFNO and SLNO are not monotonic at every intermediate difficulty, the hard-to-easy increase is substantially smaller for SLNO than for either baseline. The corresponding slow-branch errors show no comparable hard-to-easy escalation, indicating that the degradation induced by increasing $\phi_{\max}$ is concentrated in the fast oscillatory dynamics. Full values are reported in Appendix E.5.

Table 1: Controlled models at $\phi_{\max} = 1.50$. Every model has four blocks. Fast@1 and Fast@20 are mean $\pm$ sample standard deviation over three seeds, evaluated with the same branch metric and test trajectories.

Model	Width	Params	Fast@1	Fast@20
SFNO (Bonev et al., 2023)	148	364,968	$\mathbf{0.01691 \pm 0.00308}$	0.15661 ± 0.00283
GSNO (Tang et al., 2026)	61	362,218	0.02112 ± 0.00330	0.28282 ± 0.03211
SLNO	32	367,060	0.02558 ± 0.00437	$\mathbf{0.07556 \pm 0.00778}$

Table 2: Aligned native-resolution shallow-water rollout error at 256×512. Values are mean $\pm$ population standard deviation across three training-seed means; each training-seed mean pools 150 fresh initial conditions.

Model	Params	1h	5h	10h	20h
GSNO	1.488M	$9.934 \times 10^{-5} \pm 1.82 \times 10^{-6}$	$9.511 \times 10^{-4} \pm 1.47 \times 10^{-4}$	$3.480 \times 10^{-3} \pm 1.06 \times 10^{-3}$	$4.024 \times 10^{-2} \pm 4.40 \times 10^{-2}$
SFNO	1.459M	$5.972 \times 10^{-4} \pm 4.74 \times 10^{-7}$	$2.236 \times 10^{-3} \pm 8.53 \times 10^{-5}$	$5.869 \times 10^{-3} \pm 3.46 \times 10^{-3}$	$1.753 \times 10^{-1} \pm 2.40 \times 10^{-1}$
SLNO	1.460M	$6.422 \times 10^{-4} \pm 5.32 \times 10^{-6}$	$2.225 \times 10^{-3} \pm 7.44 \times 10^{-5}$	$2.349 \times 10^{-3} \pm 1.87 \times 10^{-4}$	$2.689 \times 10^{-3} \pm 1.30 \times 10^{-4}$

Short- versus long-horizon behavior. At the hardest setting, the relative ordering of the models changes substantially between the first and twentieth rollout steps (Table 1). SFNO has the lowest Fast@1 error, whereas SLNO achieves the lowest Fast@20 error. The corresponding validation one-step total errors, which use a different metric from Fast@1, are 0.01245 ± 0.00040 for SFNO, 0.01555 ± 0.00040 for GSNO, and 0.01895 ± 0.00050 for SLNO. Thus, SLNO's long-horizon advantage does not originate from a lower one-step prediction error but emerges under repeated autoregressive rollout. Together with the difficulty sweep, these results indicate that SLNO better preserves the fast temporal dynamics as predictions are repeatedly composed in this controlled system.

5.2 Long-Horizon Shallow-Water Forecasting

Metric and shared seed protocol. After reversing the stored source-training normalization, we use the joint physical relative L_2 error in Eq. (56). Quadrature-weighted squared errors are pooled over channels and grid points and compared with the quadrature-weighted squared target norm. Geopotential, vorticity, and divergence keep their physical units in this pooled score. For every native-resolution and zero-shot result below, training seeds 666, 667, and 668 are each paired with evaluation seeds 888881, 888882, and 888883 and with 50 fresh initial conditions. We pool those 150 trajectories within each training seed, then report the mean and population standard deviation of the three training-seed means. Each model and resolution therefore contributes 450 trajectories from three independently trained models. The native-resolution and zero-shot comparisons below both follow this protocol. Appendix F records the grid contract, the optimizer, and the realization details.

Native-resolution forecasting. Table 2 reverses the short- and long-horizon ranking. At 1 h the mean-error order is GSNO, then SFNO, then SLNO. At 5 h the lowest mean is GSNO; the SLNO and SFNO means differ by about 1.1×10^{-5}, which is smaller than either training-seed standard deviation. SLNO has the lowest mean at 10 h and 20 h, ahead of GSNO and then SFNO. The 10 h reductions are 32.5% relative to GSNO and 60.0% relative to SFNO; at 20 h, the reductions of the mean of the three training-seed means are 93.3% and 98.5%. Appendix Table 8 lists those three means. The largest SLNO entry remains below the smallest GSNO entry and the smallest SFNO entry. Appendix F.2 provides the corresponding seed-wise and per-trajectory error statistics. From 5 h to 20 h, SLNO's mean stays near 2.2–2.7×10^{-3}, while both baseline means increase. Appendix F.3 records the randomly selected qualitative trajectory and its corresponding rollout errors.

Zero-shot spatial-resolution transfer. The same checkpoints, trained only at 256×512, are evaluated on 96×192, 128×256, 160×320, 192×384, 224×448, and 256×512. Learned parameters and the source-training normalization stay fixed, with no retraining or fine-tuning, and each res-

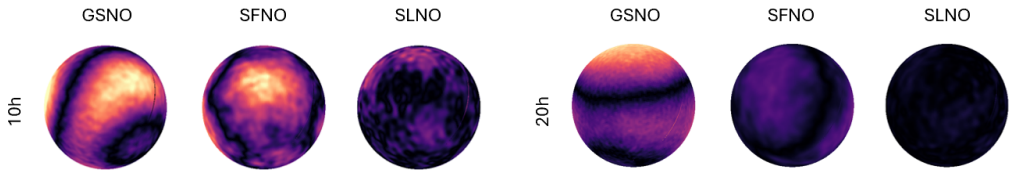

Figure 3: Long-horizon shallow-water rollout error at the native 256×512 resolution. Each sphere shows absolute height error in meters for the same randomly selected evaluation trajectory, with a shared color scale across methods within each row; darker regions indicate smaller error. The trajectory was selected independently of model prediction errors. Aggregate quantitative comparisons use the complete evaluation set.

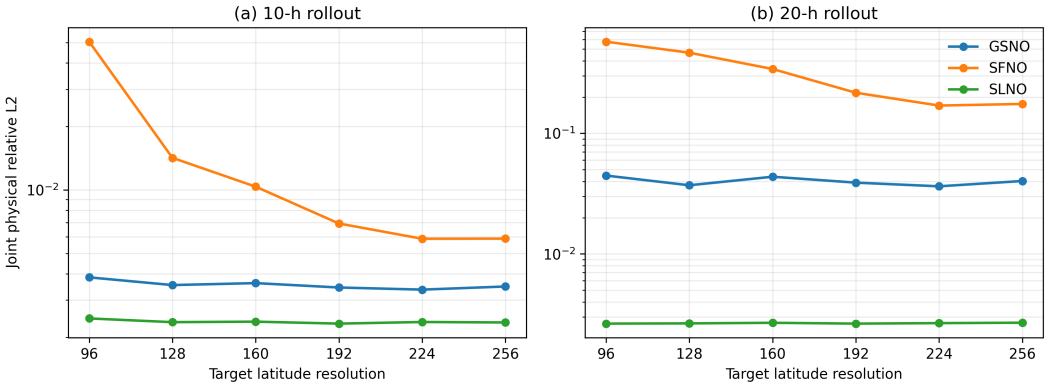

Figure 4: Zero-shot spatial-resolution transfer of checkpoints trained only at 256×512. Learned parameters remain frozen at every target resolution. The panels show 10 h and 20 h mean errors on logarithmic axes; lines connect evaluated grids. SLNO attains the lowest mean error in both panels throughout the tested range. Each target grid uses its native shallow-water initial-condition distribution, so this measures transfer across target-resolution distributions. Full mean $\pm$ population standard deviation results are in Table 10.

olution uses the training-seed protocol above. GSNO and SFNO change the external grid while retaining the source internal spectral scaffold; their weights are not interpolated or remapped, and SFNO's discrete-cosine degree basis is unchanged. SLNO regenerates $\lambda_\ell = \ell(\ell+1)$ on the target retained degrees and evaluates its frozen spectral law there. Appendix F.4 records these conventions. GSNO keeps the lowest 1 h and 5 h means at all six resolutions, and SLNO keeps the lowest 10 h and 20 h means (Figure 4). SLNO's 20 h mean lies between 2.6422×10^{-3} and 2.6888×10^{-3}: the range divided by the minimum is 1.76%, against 22.81% for GSNO and 238.69% for SFNO. From 256×512 to 96×192, the 20 h changes are $+10.99\%$ for GSNO, $+228.36\%$ for SFNO, and -1.73% for SLNO; the 10 h changes are $+10.51\%$, $+759.37\%$, and $+4.52\%$. Each target solver draws its native Mach-0.2 initial-condition distribution, so the available spectral band changes with resolution. Matched resolution labels identify target distributions, and the small SLNO range is an empirical observation on these six distributions.

5.3 Global Weather Forecasting

We next compare GSNO and SLNO on global weather forecasting. Both models use the same data, training protocol, training duration, rollout trajectories, and evaluation procedure. The input state contains 27 variables, of which the models predict six. One prediction step corresponds to a 24-hour forecast increment. The evaluation contains 512 trajectories and extends to a maximum lead time of 30 days. Appendix G lists the predicted variables and the detailed evaluation protocol.

The weather experiment uses a partial-autoregressive conditional rollout: the six predicted variables are recursively fed back, while the remaining 21 input channels at each lead time are supplied

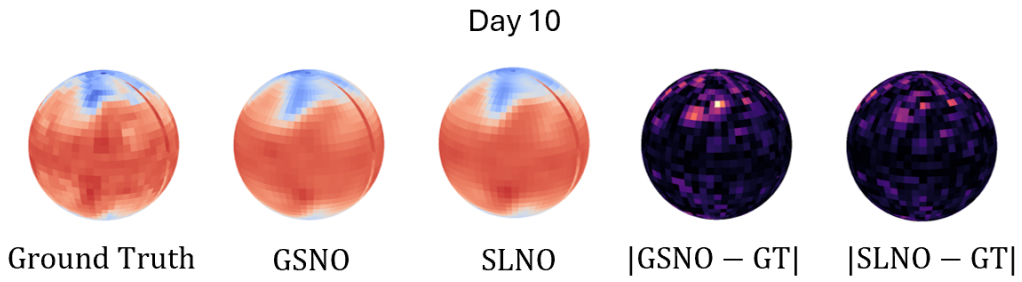

Figure 5: Day-10 temperature at 600 hPa on the same randomly selected partial-autoregressive evaluation trajectory as Appendix Figure 10. Columns show the target, GSNO, SLNO, and the corresponding absolute errors. The trajectory was selected independently of model prediction errors. Aggregate quantitative comparisons use all 512 evaluation trajectories.

Table 3: Latitude-weighted RMSE and ACC over the 512 weather trajectories. Columns denote forecast lead times in days. Lower wRMSE and higher ACC are better.

Model	Score	1	2	3	5	10	15	20	30
GSNO	wRMSE	2.886712	3.143767	3.226481	3.289301	3.382735	3.447609	3.459270	3.455369
SLNO	wRMSE	2.861572	3.100744	3.181690	3.247958	3.333658	3.385863	3.388283	3.391095
GSNO	ACC	0.840906	0.809903	0.799613	0.792474	0.785932	0.782167	0.782692	0.782854
SLNO	ACC	0.842786	0.812520	0.801833	0.793791	0.788494	0.786683	0.788296	0.787944

from the corresponding ERA5 state (Hersbach et al., 2020). Predictions and targets are evaluated in physical scale using latitude-weighted RMSE (wRMSE), with latitude weights proportional to $\cos(\text{latitude})$ and normalized to have mean one. The metric is aggregated over the six predicted variables, and ACC is reported alongside wRMSE.

Table 3 reports the partial-autoregressive rollout results at eight lead times over the 512 evaluation trajectories. Across all reported lead times, SLNO obtains lower wRMSE and higher ACC than GSNO. The relative reduction in wRMSE is approximately 2.05% at day 20 and 1.86% at day 30, indicating that the improvement is maintained throughout the long-horizon rollout. Appendix Figure 10 and Figure 5 show qualitative predictions from the same randomly selected evaluation trajectory, while all quantitative comparisons are based on the complete evaluation set.

6 Conclusion

In this work, we introduced the Spherical Laplace Neural Operator (SLNO) for learning long-term spherical dynamics through explicit spatio-temporal spectral modeling. By combining spherical harmonic representations with scale-conditioned Laplace transfer functions, SLNO extends conventional spatial spectral operators to incorporate temporal responses characterized by learnable decay and oscillatory modes. We derived analytic initial-state and trajectory-driven responses and integrated them into nonlinear neural operator blocks, enabling structured temporal modeling under single-step supervision. SLNO provides a structured framework for incorporating spatially dependent transient dynamics into spherical neural operator learning. Experiments on controlled fast–slow dynamics and spherical shallow-water equations demonstrate improved long-horizon prediction, with additional evidence of zero-shot spatial-resolution transfer. Global weather experiments further show consistent improvements over the evaluated baseline under partial-autoregressive forecasting. These results support that explicit temporal modeling provides an effective inductive bias for spherical evolution.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing, translation, identification of related work, and code development. For manuscript preparation, these tools helped refine grammar, wording, and readability, and check the internal consistency of the text. For code development, generative AI assisted in drafting and modifying parts of the model training and experiment scripts, long-horizon rollout evaluation programs, result aggregation and verification scripts, and visualization code. The authors checked the adopted code through manual inspection, execution, controlled comparisons, and verification of the resulting outputs. These checks focused on ensuring that model configurations, checkpoint loading, training and evaluation protocols, prediction variables, rollout procedures, and evaluation metrics were consistent with the experimental settings described in the paper. Issues involving paths, configurations, evaluation procedures, result reproducibility, or visualization were manually corrected, and the relevant programs were rerun to verify the corrections. The numerical results reported in this paper were obtained from the experiments described in the manuscript. The authors reviewed and confirmed the final code and experimental results and take full responsibility for the final content of the paper, including AI-assisted text, code, and other research artifacts.

REPRODUCIBILITY STATEMENT

Appendices E, F, and G document the experimental configurations, data contracts, evaluation protocols, metrics, and aggregation procedures used for the reported results. The supplementary material provides the SLNO implementation used in the global weather experiment, including the internal latent-time realization.

Supplementary Material provides the SLNO implementation used in the global weather experiment, including the internal latent-time realization.

REFERENCES

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *International conference on machine learning*, pp. 2806–2823. PMLR, 2023.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, 2024.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE transactions on neural networks*, 6(4):911–917, 1995.

Chun-Wun Cheng, Jiahao Huang, Yi Zhang, Guang Yang, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. Mamba neural operator: Who wins? transformers vs. state-space models for pdes. *Journal of Computational Physics*, pp. 114567, 2025.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pp. 1321–1330. PMLR, 2019.

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

Michaël Defferrard, Martino Milani, Frédérick Gusset, and Nathanaël Perraudin. Deepsphere: a graph-based spherical cnn. *arXiv preprint arXiv:2012.15000*, 2020.

James R Driscoll and Dennis M Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in applied mathematics*, 15(2):202–250, 1994.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive fourier neural operators. In *International conference on learning representations*, 2021.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34:24048–24062, 2021.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Nodens Koren and Samuel Lanthaler. Merging memory and space: A state space neural operator. *arXiv preprint arXiv:2507.23428*, 2025.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

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

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020b.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020c.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

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.

Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. *arXiv preprint arXiv:2204.11127*, 2022.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Hao Tang, Jiongyu Zhu, Zimeng Feng, Hao Li, and Chao Li. Geometric laplace neural operator. *arXiv preprint arXiv:2512.16409*, 2025.

Hao Tang, Hao Chen, and Chao Li. Generalized spherical neural operators: Green's function formulation. In *International Conference on Learning Representations*, volume 2026, pp. 11180–11201, 2026.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. *arXiv preprint arXiv:2111.13802*, 2021.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023.

A Detailed Operator Derivation

A.1 Spherical reduction

Assume a finite-band, channel-valued, zonal kernel with a causal temporal convolution. For each lag a, expand its spatial part as

$$\kappa(x, a) = \sum_{\ell=0}^{L} \frac{2\ell+1}{4\pi} \kappa_\ell(a) P_\ell(x), \tag{21}$$

where P_ℓ is the Legendre polynomial and κ_ℓ is a channel matrix. The addition theorem gives

$$\frac{2\ell+1}{4\pi} P_\ell(\omega \cdot \eta) = \sum_{m=-\ell}^{\ell} Y_\ell^m(\omega) \overline{Y_\ell^m(\eta)}. \tag{22}$$

Substituting into Eq. (10) and using orthonormality yields

$$\widehat{z}_{\ell m}(t) = \int_0^t \kappa_\ell(t-a) \widehat{v}_{\ell m}(a)\, da. \tag{23}$$

The same channel response therefore acts on all orders at fixed degree.

The temporal convolution is an assumption; it is not inferred from spatial equivariance alone. On a finite retained band, the spatial sums are finite. Assuming the temporal integrability conditions needed for the unilateral Laplace transform gives Eq. (11) on a common half-plane of convergence.

A.2 Shared Spectral Parameterization

For a network with D blocks, each block generates its dictionary from a normalized version of λ_ℓ. The initial block produces

$$\mathcal{D}_\ell^{(0)} = \left\{ \mu_{r,\ell}^{(0)}, A_{r,\ell}^{(0)} \right\}_{r=1}^{R},$$

while subsequent blocks produce

$$\mathcal{D}_\ell^{(q)} = \left\{ \mu_{r,\ell}^{(q)}, B_{r,\ell}^{(q)} \right\}_{r=1}^{R}, \qquad q = 1, \ldots, D-1.$$

A shared spectral feature network maps the normalized Laplace–Beltrami eigenvalue to a latent spectral representation, from which separate output heads generate the pole locations and response coefficients. These mappings are shared across spherical degrees within each block, defining

$$\mathcal{D}_\ell^{(q)} = g_{\theta_q}(\lambda_\ell), \qquad q = 0, \ldots, D-1.$$

Different blocks have independent parameters. Sharing is imposed by g_{θ_q}, rather than by the one-to-one relabeling of ℓ as λ_ℓ. Non-real poles and responses are paired with their complex conjugates, and real poles use real responses.

A.3 Pole–residue kernel

The rational transfer in Eq. (12) has causal kernel

$$\kappa_\ell(t) = \sum_{r=1}^{R} B_{r,\ell} e^{\mu_{r,\ell} t}, \qquad t \geq 0. \tag{24}$$

Poles are simple in this representation. Repeated locations combine their residues and do not introduce higher-order poles. The response matrices need not be diagonal or low matrix rank.

B FINITE TEMPORAL REALIZATION

B.1 INITIAL RESPONSE AND LATENT TIME

The initial block uses Eq. (13), with its own dictionary independent of subsequent blocks. A literal physical state propagator would require $\Phi_\ell^{(0)}(0) = I$, equivalently

$$\sum_r A_{r,\ell}^{(0)} = I. \tag{25}$$

A latent input–output response need not satisfy this normalization, and the evaluated shallow-water model does not enforce it.

The retained nodes are $t_n = n\delta t$, $n = 0, \dots, N_t - 1$. Their endpoint and DFT period are

$$T = (N_t - 1)\delta t, \qquad T_{\mathrm{DFT}} = N_t \delta t. \tag{26}$$

The retained nodes define an internal latent temporal coordinate used to construct the SLNO response. The network decoder associates the endpoint of this latent trajectory with the outer physical forecast horizon.

B.2 ANALYTIC TRAJECTORY RESPONSE

The frequencies use signed DFT ordering, $\omega_k = 2\pi k/(N_t\delta t)$, with the bin index read on $[-\lfloor N_t/2 \rfloor, \dots, \lfloor (N_t - 1)/2 \rfloor]$. The evaluated configuration retains all modes with odd N_t; its DFT period $N_t\delta t$ differs from T.

For odd N_t, continue the samples by the finite interpolant

$$\widetilde{v}_{\ell m}(t) = \sum_k \alpha_{k,\ell m} e^{i\omega_k t}. \tag{27}$$

This agrees with the retained samples at their nodes. Its unilateral Laplace transform is

$$V_{\ell m}^{\mathrm{mer}}(s) = \sum_k \frac{\alpha_{k,\ell m}}{s - i\omega_k}. \tag{28}$$

This is the transform of the continued interpolant, not of a trajectory zero-extended beyond T.

Substitution into Eq. (23) gives

$$\widehat{z}_{\ell m}^{(f)}(t) = \sum_{r,k} B_{r,\ell}\alpha_{k,\ell m} \int_0^t e^{\mu_{r,\ell}(t-a)} e^{i\omega_k a}\, da \tag{29}$$

$$= \sum_{r,k} B_{r,\ell}\alpha_{k,\ell m}\psi_t(\mu_{r,\ell}, i\omega_k). \tag{30}$$

This establishes Eq. (16) exactly for the finite trigonometric interpolant defined above, which is the temporal representation used by SLNO.

B.3 COLLISION-SAFE EVALUATION

Define

$$\mathrm{exprel}(z) = \frac{e^z - 1}{z}, \qquad \mathrm{exprel}(0) = 1. \tag{31}$$

Then

$$\psi_t(\mu, \xi) = te^{\xi t}\,\mathrm{exprel}((\mu - \xi)t). \tag{32}$$

The implementation uses a Taylor expansion near zero and an exponential-minus-one evaluation elsewhere. At a collision the response is $te^{\mu t}$, avoiding division by zero or subtraction of nearly equal exponentials.

B.4 Conjugation and the even-length Nyquist mode

Let $\bar{k} = (-k) \bmod N_t$. Real spherical fields satisfy

$$\alpha_{k,\ell,-m} = (-1)^m \overline{\alpha_{\bar{k},\ell,m}}. \tag{33}$$

Any retained temporal subset must be closed under $k \mapsto \bar{k}$.

For even N_t, the Nyquist samples admit the real continuation $\cos(\Omega_{\mathrm{Ny}} t)$, where $\Omega_{\mathrm{Ny}} = \pi/\delta t$. Define

$$\vartheta_t(\mu; k) = \begin{cases} \psi_t(\mu, i\omega_k), & k \neq N_t/2, \\ \frac{1}{2}\left[\psi_t(\mu, i\Omega_{\mathrm{Ny}}) + \psi_t(\mu, -i\Omega_{\mathrm{Ny}})\right], & k = N_t/2 \text{ for even } N_t. \end{cases} \tag{34}$$

For odd N_t, only the first case applies. When using an even grid, replace the excitation response by ϑ_t in the finite reconstruction.

Conjugate pairing of poles and matrices, real self-conjugate branches, and this excitation closure imply

$$\widehat{z}_{\ell,-m}(t) = (-1)^m \overline{\widehat{z}_{\ell,m}(t)}. \tag{35}$$

Inverse SHT therefore yields a real field up to numerical roundoff. Real-valued reconstruction therefore uses conjugate pairing of poles and matrices together with an excitation set closed under $k \mapsto \bar{k}$.

B.5 Latent-window evaluation

The temporal Fourier coefficients are formed from the complete retained latent trajectory within each block. These samples are generated internally by SLNO and are not observed future states. The finite implementation therefore evaluates the temporal response over an internal latent window, while the underlying continuous convolution kernel retains its causal form.

For the final block, only the endpoint at T needs to be evaluated when subsequent operations are deterministic and time-separable. Intermediate blocks retain the full latent trajectory because the following trajectory block constructs its temporal coefficients from that retained window.

C Representation Properties and Interpretation

Fixed-step relation. For a homogeneous linear response, setting

$$W_\ell = \Phi_\ell(\Delta) \tag{36}$$

reproduces the same endpoint through a degree-wise multiplier. This relation highlights that the distinction of SLNO lies in its explicit temporal factorization of the modal response rather than in the representation of a single fixed-step endpoint alone. On a retained spherical band, ℓ and $\lambda_\ell = \ell(\ell+1)$ contain the same discrete degree information, while λ_ℓ provides a natural spectral coordinate for parameterizing the shared response law.

Interpretation of learned poles. Even in the scalar one-pole homogeneous case, a single endpoint admits

$$\mu_q = \mu + \frac{2\pi i q}{\Delta}, \qquad e^{\mu_q \Delta} = e^{\mu \Delta}, \qquad q \in \mathbb{Z}. \tag{37}$$

For the internal latent clock, the analogous relation is defined with respect to its endpoint T. With learned residues and encoders/decoders, multiple latent parameterizations can therefore realize the same supervised endpoint behavior. Accordingly, under single-step endpoint supervision, the learned poles are naturally interpreted as parameters of the latent temporal representation rather than as uniquely identified physical PDE poles.

State-space realization. A finite simple-pole transfer admits branch states

$$\dot{x}_{r,\ell m} = \mu_{r,\ell} x_{r,\ell m} + B_{r,\ell} \widehat{v}_{\ell m}, \qquad \widehat{z}^{(f)}_{\ell m} = \sum_r x_{r,\ell m}. \tag{38}$$

For a homogeneous branch, $x_{r,\ell m}(0) = A^{(0)}_{r,\ell} \widehat{h}_{\ell m}$ produces the corresponding initial response. Thus each rational temporal branch has an explicit finite-dimensional state-space realization. Across nonlinear SLNO blocks, these branch states and pole dictionaries serve as block-specific latent representations.

Symmetry and nonlinear coupling. The zonal equivariant linear branch yields an order-independent modal response within each spherical degree. More general anisotropic response laws can be represented by extending the parameterization to include order dependence. Between linear spectral responses, pointwise nonlinearities couple spherical harmonic degrees, allowing the full network to model nonlinear interactions beyond independent linear modal evolution.

D Amplification and Rollout Analysis

D.1 Finite-axis and endpoint gains

For a retained trajectory define

$$\|v\|_{\mathsf{T}}^2 = \frac{1}{N_t} \sum_n \|v_n\|_2^2. \tag{39}$$

The forward-normalized DFT satisfies

$$\sum_k \|\alpha_k\|_2^2 = \|v\|_{\mathsf{T}}^2. \tag{40}$$

At fixed degree, form

$$\mathbf{R}_\ell = [R_{\ell,nk}]_{n,k}, \qquad R_{\ell,nk} = \sum_r B_{r,\ell} \vartheta_{t_n}(\mu_{r,\ell}; k), \tag{41}$$

and

$$\mathbf{P}_\ell = \begin{bmatrix} \Phi_\ell(t_0) \\ \vdots \\ \Phi_\ell(t_{N_t-1}) \end{bmatrix}. \tag{42}$$

Their exact induced gains in these coefficient norms are

$$\gamma_{\ell,\mathrm{full}}^{(f)} = N_t^{-1/2} \sigma_{\max}(\mathbf{R}_\ell), \qquad \gamma_{\ell,\mathrm{full}}^{(0)} = N_t^{-1/2} \sigma_{\max}(\mathbf{P}_\ell). \tag{43}$$

For a hypothetical joint response with product input norm $\|h\|_2^2 + \|v\|_{\mathsf{T}}^2$, the gain is

$$N_t^{-1/2} \sigma_{\max}([\mathbf{P}_\ell \ \mathbf{R}_\ell]). \tag{44}$$

The implemented blocks activate their respective initial or driven branch. Endpoint gains use the last block row without the output factor $N_t^{-1/2}$.

These statements hold on the complex coefficient space. Under the real-closure conditions of Appendix B, the response commutes with the conjugation involution. Decomposing a complex vector as $a + ib$ with real-subspace vectors a, b shows that the complexified norm equals its restriction to admissible real fields.

A sphere-wide gain is the maximum over retained degrees in the orthonormal coefficient metric. Transfer to a weighted grid norm additionally requires $S^* Q S = I$, where S is synthesis and Q contains quadrature weights. No identity with an arbitrary unweighted tensor norm is asserted.

D.2 Stable poles and coordinate dependence

Multiplying residues by a scalar changes gain without changing poles. Left-half-plane poles therefore do not imply contraction. Gains also depend on latent coordinates. Synchronized transformations

$$E' = TE, \qquad K' = TKT^{-1}, \qquad D' = DT^{-1}$$

preserve DKE but can change Euclidean singular values when T is non-unitary. Internal gains are fixed-coordinate diagnostics, not intrinsic physical stability certificates.

D.3 NONLINEAR COMPOSITION

For a block $B_j(v) = \Psi_j(K_j v, R_j v)$ with separate Lipschitz constants a_j, b_j, the triangle inequality gives

$$\mathrm{Lip}(B_j) \le a_j\|K_j\| + b_j\|R_j\|. \tag{45}$$

For target and learned maps G_Δ, F_Δ, set

$$x_{n+1} = G_\Delta(x_n), \qquad \widehat{x}_{n+1} = F_\Delta(\widehat{x}_n),$$
$$e_n = \widehat{x}_n - x_n, \qquad \epsilon_n = \|F_\Delta(x_n) - G_\Delta(x_n)\|.$$

If the derivative is bounded on the joining segment, let

$$\rho_n = \sup_{a\in[0,1]} \|DF_\Delta(x_n + a e_n)\|. \tag{46}$$

Then

$$\|e_{n+1}\| \le \rho_n\|e_n\| + \epsilon_n, \tag{47}$$

$$\|e_n\| \le \left(\prod_{j=0}^{n-1} \rho_j\right)\|e_0\| + \sum_{k=0}^{n-1}\left(\prod_{j=k+1}^{n-1} \rho_j\right)\epsilon_k. \tag{48}$$

This conditional inequality explains the distinction between one-step defect and error amplification. It does not show that training produces contractive maps or certify the evaluated checkpoints.

E CONTROLLED FAST–SLOW EXPERIMENT: ADDITIONAL DETAILS

E.1 CONTROLLED EQUATIONS AND MODAL DYNAMICS

The state is $q = (\zeta, d, h)$, with

$$\begin{aligned} \partial_t\zeta &= -\gamma_s\zeta, \\ \partial_t d &= -\gamma_f d + c^2\Delta_{\mathbb{S}^2} h, \\ \partial_t h &= d - \gamma_f h. \end{aligned} \tag{49}$$

Spherical harmonics satisfy $-\Delta_{\mathbb{S}^2} Y_\ell^m = \lambda_\ell Y_\ell^m$ with $\lambda_\ell = \ell(\ell+1)$. Every retained degree therefore carries

$$\mu_{\mathrm{slow}} = -\gamma_s, \qquad \mu_{\mathrm{fast},\pm,\ell} = -\gamma_f \pm \mathrm{i}\, c\sqrt{\lambda_\ell}. \tag{50}$$

The slow and fast branches are not two disjoint spatial bands.

Degrees $1 \le \ell \le 15$ are retained on a 32×64 equiangular grid, with $\gamma_s = 0.05$, $\gamma_f = 0.10$, and $\Delta t = 0.06$.

The state is sampled in real spherical coefficients. Gaussian amplitudes scale as $(2\ell+1)^{-1/2}$, with an additional $1/(c\sqrt{\lambda_\ell})$ scaling for h. Each sample is rescaled to equal slow and fast energies. The slow and fast spectral norms used for this balance and for evaluation are

$$\|q\|_s^2 = \sum_{\ell,m} |\zeta_{\ell m}|^2, \qquad \|q\|_f^2 = \sum_{\ell,m} \left(|d_{\ell m}|^2 + \omega_\ell^2 |h_{\ell m}|^2\right), \tag{51}$$

where $\omega_\ell = c\sqrt{\lambda_\ell}$.

E.2 EXACT-REFERENCE GENERATION

Reference trajectories are generated by exact modal propagation. For $\omega_\ell = c\sqrt{\lambda_\ell}$ and any interval τ,

$$\zeta_{\ell m}(t+\tau) = e^{-\gamma_s\tau}\zeta_{\ell m}(t), \tag{52}$$

$$\begin{pmatrix} d_{\ell m}(t+\tau) \\ h_{\ell m}(t+\tau) \end{pmatrix} = e^{-\gamma_f\tau}\begin{pmatrix} \cos(\omega_\ell\tau) & -\omega_\ell\sin(\omega_\ell\tau) \\ \sin(\omega_\ell\tau)/\omega_\ell & \cos(\omega_\ell\tau) \end{pmatrix}\begin{pmatrix} d_{\ell m}(t) \\ h_{\ell m}(t) \end{pmatrix}. \tag{53}$$

These expressions generate both the one-step targets and the reference rollouts for the prescribed linear dynamics.

Table 4: Controlled step-20 fast-branch relative error. Entries are mean ± sample standard deviation over three matched seeds.

$\phi_{\max}$	SFNO	GSNO	SLNO
0.25	0.08736 ± 0.00637	0.12295 ± 0.00223	$\mathbf{0.06505 \pm 0.00637}$
0.50	0.11017 ± 0.00440	0.18065 ± 0.00908	$\mathbf{0.06065 \pm 0.00534}$
1.00	0.16481 ± 0.00824	0.25970 ± 0.01926	$\mathbf{0.07736 \pm 0.00524}$
1.50	0.15661 ± 0.00283	0.28282 ± 0.03211	$\mathbf{0.07556 \pm 0.00778}$

E.3 Dataset and optimization protocol

Training and validation source states are sampled at stratified trajectory steps 0–49, rebalanced, and propagated analytically for one target step. This trajectory-state sampling produces 1,536 training pairs and 384 validation pairs. Training-derived channel scales normalize the network interface. Test trajectories start from independently sampled, energy-balanced initial states, with 12 test initial conditions per seed. Validation one-step error and test Fast@1 are evaluated on different data and should therefore be interpreted as distinct quantities.

Each of seeds 0, 1, and 2 uses 40 epochs, batch size 8, AdamW (Loshchilov & Hutter, 2017) with learning rate 10^{-3} and weight decay 10^{-5}, cosine annealing, and gradient clipping at 1.0. The checkpoint with the best one-step validation error is evaluated by native-grid autoregressive rollout. Rollouts are recorded through step 50, with step 20 used as the primary long-horizon comparison point in the main text. Three seeds, four values of $\phi_{\max}$, and three models give 36 runs.

E.4 Matched operator configurations

Widths, blocks, and parameter counts are reported in Table 1. Widths remain fixed across phase-difficulty settings and seeds. SLNO uses 25 retained temporal nodes, one real pole and two complex-conjugate pole pairs, spectral hidden width 16, unconstrained pole real parts, and pole time scale 0.36.

All models are evaluated within the same matched block family. The SFNO and GSNO configurations use their official global-operator bodies verbatim within this common host, allowing the comparison to isolate differences between the global operators under the same surrounding architecture and training protocol.

E.5 Complete fast-branch statistics

The fast relative error on a trajectory is

$$E_{\text{fast}} = \left(\frac{\sum_{\ell,m} \left(|\hat{d}_{\ell m} - d_{\ell m}|^2 + \omega_\ell^2 |\hat{h}_{\ell m} - h_{\ell m}|^2 \right)}{\sum_{\ell,m} \left(|d_{\ell m}|^2 + \omega_\ell^2 |h_{\ell m}|^2 \right)} \right)^{1/2}. \tag{54}$$

The slow relative error is the analogous ratio on ζ. Denominators are floored at 10^{-12} before taking the square root. Each run mean averages the 12 test trajectories.

For three run means a_i, the sample standard deviation is

$$s = \left[\frac{\sum_{i=1}^{3} (a_i - \bar{a})^2}{2} \right]^{1/2}. \tag{55}$$

Table 4 gives the values plotted in Figure 2(a). The paper reports the mean of the three run-level means together with this sample standard deviation. The same convention is used throughout the controlled experiment.

From $\phi_{\max} = 0.25$ to 1.50, the hard-to-easy ratios of the mean Fast@20 errors are approximately 1.793 for SFNO, 2.300 for GSNO, and 1.162 for SLNO. Thus, both baselines show substantially larger increases across the difficulty sweep than SLNO.

Table 5: Controlled step-20 slow-branch relative error. Entries are mean ± sample standard deviation over three matched seeds.

$\phi_{\max}$	SFNO	GSNO	SLNO
0.25	0.02208 ± 0.00169	0.03263 ± 0.00752	$\mathbf{0.01546 \pm 0.00131}$
0.50	0.02029 ± 0.00226	0.03285 ± 0.00700	$\mathbf{0.01351 \pm 0.00149}$
1.00	0.02264 ± 0.00172	0.02759 ± 0.00466	$\mathbf{0.01327 \pm 0.00055}$
1.50	0.02468 ± 0.00173	0.02419 ± 0.00389	$\mathbf{0.01491 \pm 0.00104}$

Table 6: Hard-to-easy step-20 error ratios in the controlled experiment. Ratios compare $\phi_{\max} = 1.50$ with $\phi_{\max} = 0.25$.

Metric	SFNO	GSNO	SLNO
Fast@20 hard/easy	1.793	2.300	**1.162**
Slow@20 hard/easy	1.117	0.742	0.964

E.6 Complete slow-branch statistics

Table 5 gives the slow-branch values plotted in Figure 2(b).

The corresponding hard-to-easy ratios are approximately 1.117 for SFNO, 0.742 for GSNO, and 0.964 for SLNO. The slow branch therefore shows no escalation comparable to that observed in the fast branch of SFNO and GSNO. This contrast supports the interpretation that increasing $\phi_{\max}$ primarily stresses the fast oscillatory dynamics.

E.7 One-step and rollout metric distinction

Fast@1 and Fast@20 in Table 1 both use Eq. (54) on the test trajectories. At $\phi_{\max} = 1.50$, the validation one-step total errors are $0.01245328 \pm 0.00039753$ for SFNO, $0.01555412 \pm 0.00040127$ for GSNO, and $0.01894584 \pm 0.00050269$ for SLNO. The validation total error is distinct from Fast@1 because it aggregates the complete state rather than only the fast branch and is evaluated on the validation set.

At the hardest setting, SFNO has the lowest Fast@1 and the lowest validation one-step total error, whereas SLNO has the lowest Fast@20. The long-horizon advantage of SLNO therefore emerges during repeated rollout rather than from superior one-step fitting.

E.8 Difficulty-sweep ratios

For a compact comparison of sensitivity to the phase-difficulty sweep, Table 6 reports the ratio between the hardest and easiest settings using the three-seed mean errors.

The fast-branch ratio is substantially smaller for SLNO than for either baseline, whereas the slow-branch ratios remain near or below one across the three models. These statistics provide the branch-specific comparison underlying the difficulty-sweep discussion in the main text.

E.9 Qualitative sample-selection protocol

The trajectory in Figure 6 is a randomly selected qualitative example at $\phi_{\max} = 1.50$. The sample corresponds to seed 0 and test initial condition 5 and was selected independently of the prediction errors of SFNO, GSNO, and SLNO. All three learned models and the exact reference are evaluated from the same initial condition. For this trajectory, the step-20 fast-branch errors of SFNO, GSNO, and SLNO are 0.1373617, 0.3458016, and 0.0382766, respectively. The example is used only for qualitative visualization; aggregate quantitative conclusions are based on the complete evaluation set.

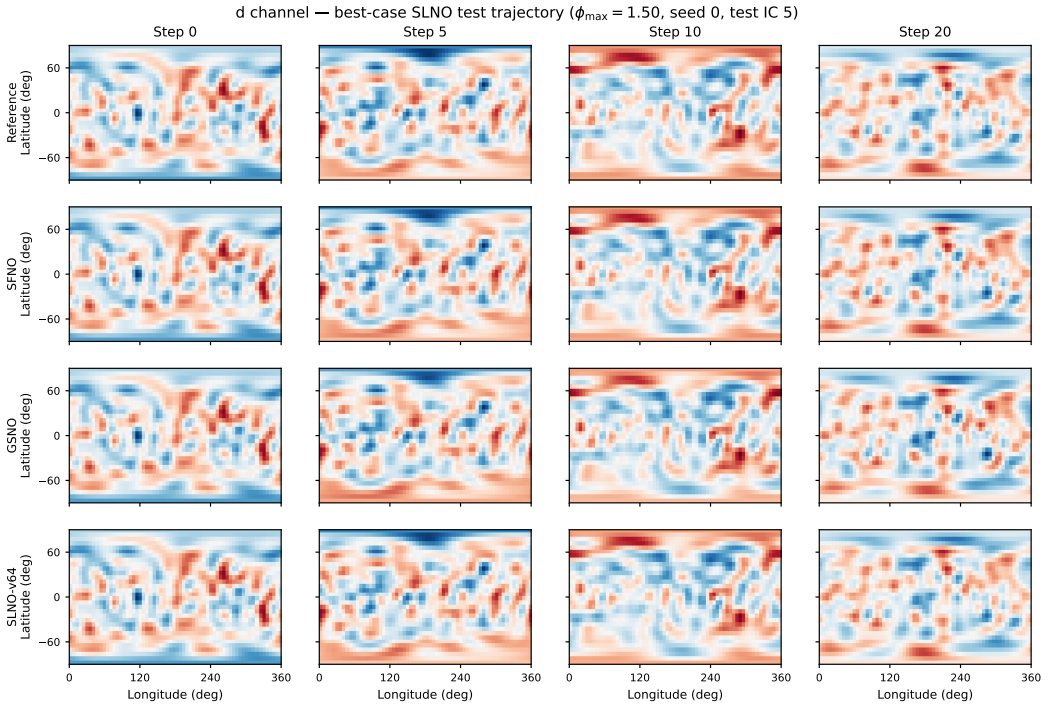

Figure 6: Randomly selected qualitative rollout of the fast-branch field d at the hardest controlled setting ($\phi_{\max} = 1.50$). Reference, SFNO, GSNO, and SLNO are evaluated from exactly the same initial condition. Within each rollout step, all rows use the same symmetric color scale. The example is used for qualitative visualization; aggregate quantitative conclusions use all three seeds and all test trajectories.

E.10 Additional qualitative results

Figure 7 shows the height channel of the same randomly selected trajectory. Figure 8 shows the fast-branch error over its rollout. Both figures use the same initial condition as Figure 6 and are intended as qualitative examples rather than aggregate estimates.

F Shallow-Water Protocol and Additional Statistics

F.1 Training and model realization

Task and grid contract. We evaluate nonlinear shallow-water forecasting on a rotating sphere (Bonev et al., 2023; Tang et al., 2026). The state comprises geopotential, vorticity, and divergence. A spectral reference solver advances randomized initial states by 24 steps of 150 seconds to form one-hour prediction targets. Training and all reported evaluations use an equiangular external grid; Legendre–Gauss coordinates may be used internally. The native resolution is 256×512. Long forecasts recursively apply the same one-hour predictor, and every reported number uses the aligned equiangular evaluation.

Models and optimization. GSNO, SFNO, and SLNO use four blocks, 64 latent channels, and spatial scale factor three. Each model is trained for 100 epochs with 256 samples per epoch, batch size one, and Adam (Kingma & Ba, 2014) with learning rate 2×10^{-3}, zero weight decay, and no scheduler. Training uses one-step endpoint targets only, with no intermediate-time supervision or autoregressive fine-tuning. We evaluate the final-epoch checkpoint. The three models are configured at comparable parameter scale, with exact recorded parameter counts reported in Table 7.

The models share the training setup in Section 5.2. Solver initial conditions use Mach parameter 0.2. Dataset initialization and normalization use seed 666; epoch-dependent training and validation

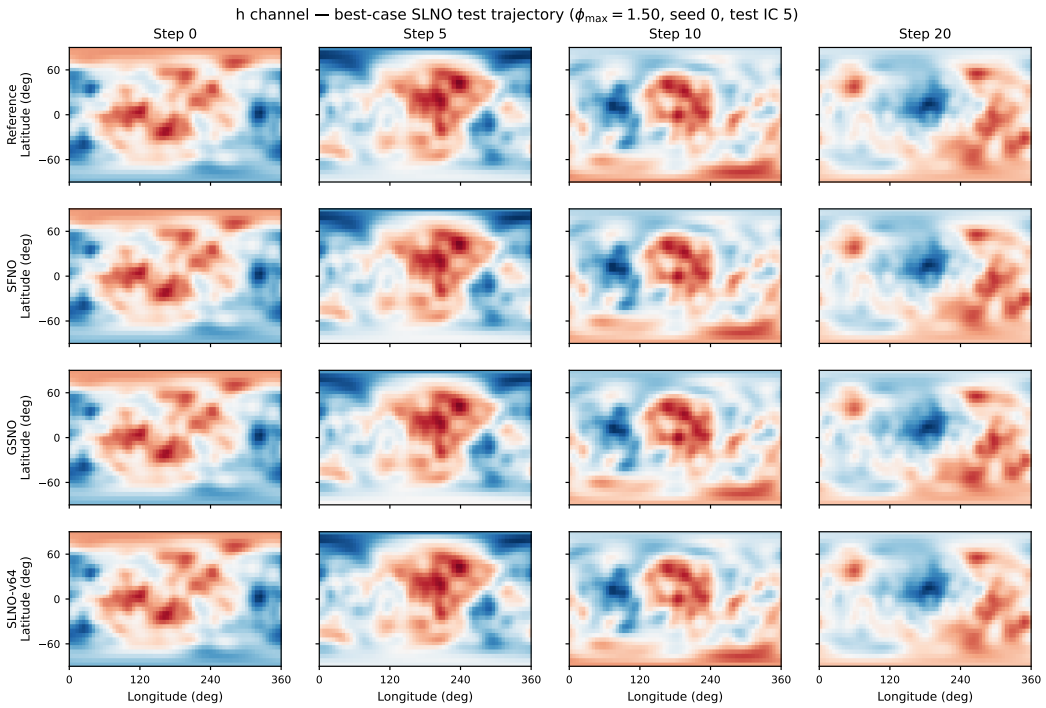

Figure 7: Height channel h for the same randomly selected trajectory as Figure 6 ($\phi_{\max} = 1.50$, seed 0, test initial condition 5). Within each rollout step, all rows share one symmetric color scale.

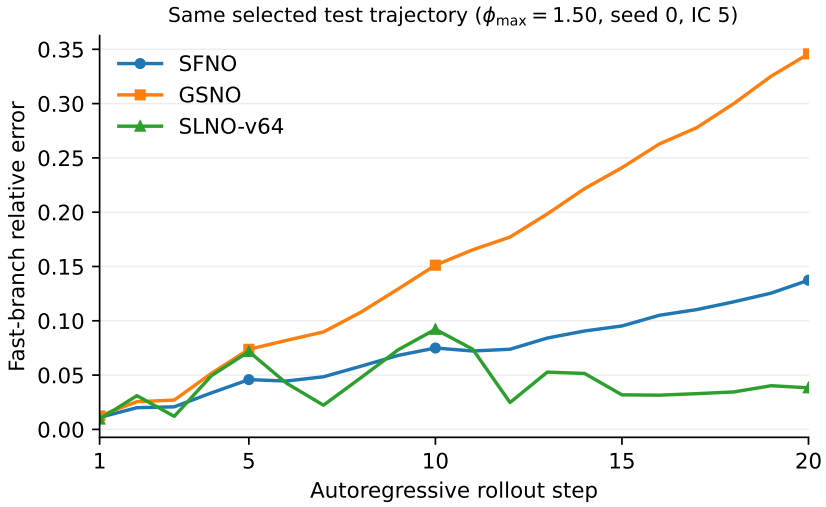

Figure 8: Fast-branch relative error on the same randomly selected trajectory. The curves correspond to one initial condition and are distinct from the three-seed aggregate statistics in Table 4.

data seeds are $100000 + e$ and $200000 + e$, respectively. Each epoch uses 256 training samples and eight validation samples. Channelwise spatial means and variances from dataset initialization define the saved normalization. Training minimizes absolute squared spherical L_2 error on normalized fields, while validation uses squared relative L_2. Reported physical rollout errors reverse the stored

Table 7: Exact recorded parameter counts for the aligned shallow-water comparison.

Model	Parameters
GSNO	1,487,981
SFNO	1,459,081
SLNO	1,459,749

Table 8: Native 256×512 rollout error at 20 h by training seed. Each entry averages 150 fresh initial conditions.

Model	666	667	668
GSNO	0.10229133	0.00513245	0.01330573
SFNO	0.00471279	0.51512948	0.00615506
SLNO	0.00264035	0.00255944	0.00286668

source-run normalization and use

$$e(\widehat{u}, u) = \left(\frac{\sum_{c,p} w_p |\widehat{u}_{c,p} - u_{c,p}|^2}{\max\left\{ \sum_{c,p} w_p |u_{c,p}|^2, 10^{-12} \right\}} \right)^{1/2}, \tag{56}$$

where w_p are spherical quadrature weights and the sum over channels pools them before the relative norm.

SLNO has four blocks, 64 latent channels, and $N_t = 25$ temporal nodes with $\delta t = 2.5$ and $T = 60$ in latent units. These latent units are distinct from the one-hour physical forecast interval. One real pole and two conjugate pairs give $R = 5$. Pole and response feature branches have hidden width 16. Pole real parts are unconstrained, and initial identity is not enforced. The first block activates the initial response; later blocks activate the trajectory-driven response.

SFNO parameterizes degree-wise multipliers as

$$W_\ell = \sum_{j=1}^{85} Q_{\ell j} B_j, \tag{57}$$

where Q is a fixed orthonormal degree basis and B_j are learned channel matrices. The full-rank basis retains the degree-wise multiplier space. Table 7 reports the exact recorded trainable parameter counts for the aligned comparison.

F.2 Aggregation and native-resolution distributions

Let e_{ijh} denote the physical relative error of trajectory j for training seed i at horizon h. At each resolution we compute

$$a_{ih} = \frac{1}{150} \sum_{j=1}^{150} e_{ijh}, \qquad \bar{a}_h = \frac{1}{3} \sum_{i=1}^{3} a_{ih}, \qquad \sigma_h = \left[\frac{1}{3} \sum_{i=1}^{3} (a_{ih} - \bar{a}_h)^2 \right]^{1/2}. \tag{58}$$

The shallow-water tables report the mean and population standard deviation across the three training-seed means. Evaluation seeds generate matched initial-condition sets across models at a given resolution.

Table 8 reports the three 20 h training-seed means. Across these runs, SLNO varies from 2.559×10^{-3} to 2.867×10^{-3}. Its largest training-seed mean is lower than the smallest corresponding mean of GSNO and SFNO. The medians of the three training-seed means are 1.331×10^{-2} for GSNO, 6.155×10^{-3} for SFNO, and 2.640×10^{-3} for SLNO, corresponding to SLNO reductions of 80.2% and 57.1% relative to GSNO and SFNO, respectively. Table 9 provides the corresponding per-trajectory distribution summaries.

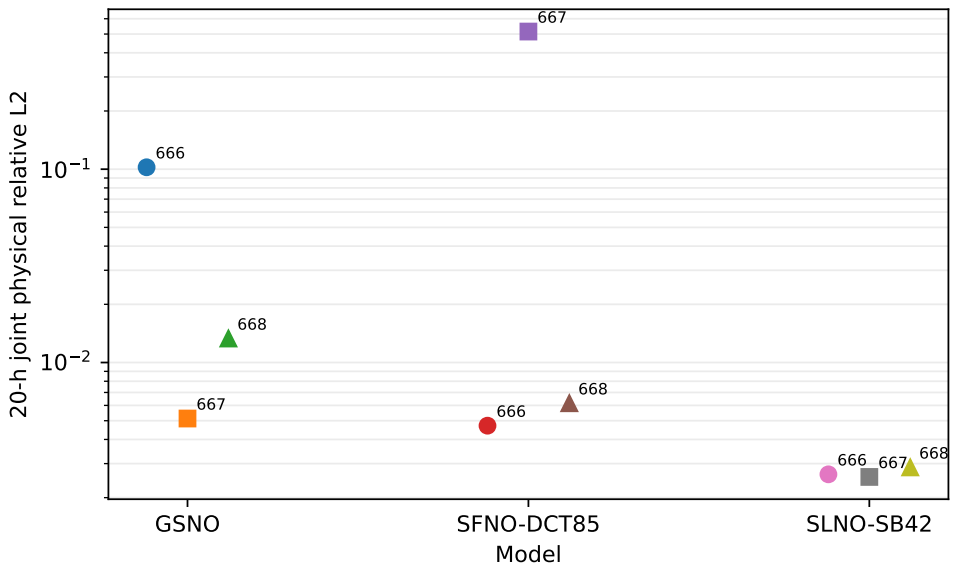

Figure 9: Training-seed variation of the 20 h joint physical relative L_2 error at the native 256×512 resolution. Each marker corresponds to one training seed (666, 667, or 668) and shows that run's mean over 150 fresh initial conditions. SLNO remains between 2.56×10^{-3} and 2.87×10^{-3} across the three runs. The largest observed SLNO training-seed mean remains below the smallest observed training-seed mean of either baseline.

Table 9: Aligned native 20 h per-trajectory error summaries in units of 10^{-3}. Each row contains 150 initial conditions; P90 denotes the 90th percentile.

Model	Seed	Mean	Median	P90	Max
GSNO	666	102.2913	49.0531	277.6892	530.6259
GSNO	667	5.1325	5.0409	6.6925	9.1770
GSNO	668	13.3057	9.4141	28.7572	69.7529
SFNO	666	4.7128	4.7269	5.2659	6.0776
SFNO	667	515.1295	370.9821	1133.2975	1857.0663
SFNO	668	6.1551	6.1372	7.4618	10.2030
SLNO	666	2.6403	2.6147	3.0626	3.4633
SLNO	667	2.5594	2.5314	2.9146	3.3738
SLNO	668	2.8667	2.8325	3.3557	3.7809

F.3 Qualitative trajectory

The qualitative example in Figure 3 is a randomly selected trajectory from the aligned native-resolution evaluation set. The trajectory was selected independently of the prediction errors of GSNO, SFNO, and SLNO. All three models are evaluated on exactly the same initial condition and under the same rollout protocol.

The displayed trajectory corresponds to training seed 668, evaluation seed 888881, and zero-based initial-condition index 5. The joint physical relative L_2 errors are 6.58×10^{-3}, 6.01×10^{-3}, and 3.00×10^{-3} at 10 h, and 2.87×10^{-2}, 7.84×10^{-3}, and 3.00×10^{-3} at 20 h, for GSNO, SFNO, and SLNO, respectively.

This trajectory is used only for qualitative visualization of the spatial error structure. All aggregate quantitative comparisons use the complete evaluation set of 450 trajectories per model.

Table 10: Full zero-shot spatial-resolution shallow-water evaluation. Values are mean ± population standard deviation across three training-seed means. Each training-seed mean pools three evaluation seeds with 50 fresh initial conditions each.

Target resolution	Model	1h	5h	10h	20h
96×192	GSNO	$2.655 \times 10^{-4} \pm 1.09 \times 10^{-5}$	$1.247 \times 10^{-3} \pm 1.25 \times 10^{-4}$	$3.846 \times 10^{-3} \pm 1.12 \times 10^{-3}$	$4.467 \times 10^{-2} \pm 4.84 \times 10^{-2}$
	SFNO	$5.068 \times 10^{-3} \pm 3.75 \times 10^{-3}$	$2.131 \times 10^{-2} \pm 1.96 \times 10^{-2}$	$5.044 \times 10^{-2} \pm 5.61 \times 10^{-2}$	$5.757 \times 10^{-1} \pm 7.91 \times 10^{-1}$
	SLNO	$6.600 \times 10^{-4} \pm 4.69 \times 10^{-6}$	$2.332 \times 10^{-3} \pm 5.25 \times 10^{-5}$	$2.456 \times 10^{-3} \pm 1.38 \times 10^{-4}$	$2.642 \times 10^{-3} \pm 8.64 \times 10^{-5}$
128×256	GSNO	$2.136 \times 10^{-4} \pm 3.12 \times 10^{-6}$	$1.222 \times 10^{-3} \pm 1.10 \times 10^{-4}$	$3.535 \times 10^{-3} \pm 9.15 \times 10^{-4}$	$3.710 \times 10^{-2} \pm 3.92 \times 10^{-2}$
	SFNO	$6.580 \times 10^{-4} \pm 4.99 \times 10^{-5}$	$2.982 \times 10^{-3} \pm 5.56 \times 10^{-4}$	$1.419 \times 10^{-2} \pm 1.48 \times 10^{-2}$	$4.664 \times 10^{-1} \pm 6.52 \times 10^{-1}$
	SLNO	$7.740 \times 10^{-4} \pm 3.25 \times 10^{-6}$	$2.329 \times 10^{-3} \pm 6.58 \times 10^{-5}$	$2.358 \times 10^{-3} \pm 1.64 \times 10^{-4}$	$2.654 \times 10^{-3} \pm 1.06 \times 10^{-4}$
160×320	GSNO	$3.646 \times 10^{-4} \pm 4.36 \times 10^{-6}$	$1.123 \times 10^{-3} \pm 1.32 \times 10^{-4}$	$3.611 \times 10^{-3} \pm 1.13 \times 10^{-3}$	$4.367 \times 10^{-2} \pm 4.82 \times 10^{-2}$
	SFNO	$6.753 \times 10^{-4} \pm 2.85 \times 10^{-6}$	$2.498 \times 10^{-3} \pm 2.30 \times 10^{-4}$	$1.035 \times 10^{-2} \pm 9.36 \times 10^{-3}$	$3.417 \times 10^{-1} \pm 4.76 \times 10^{-1}$
	SLNO	$8.349 \times 10^{-4} \pm 3.08 \times 10^{-6}$	$2.300 \times 10^{-3} \pm 7.57 \times 10^{-5}$	$2.370 \times 10^{-3} \pm 1.85 \times 10^{-4}$	$2.684 \times 10^{-3} \pm 1.37 \times 10^{-4}$
192×384	GSNO	$2.517 \times 10^{-4} \pm 5.00 \times 10^{-6}$	$1.003 \times 10^{-3} \pm 1.28 \times 10^{-4}$	$3.442 \times 10^{-3} \pm 9.83 \times 10^{-4}$	$3.901 \times 10^{-2} \pm 4.28 \times 10^{-2}$
	SFNO	$6.253 \times 10^{-4} \pm 1.97 \times 10^{-6}$	$2.260 \times 10^{-3} \pm 1.26 \times 10^{-4}$	$6.924 \times 10^{-3} \pm 4.85 \times 10^{-3}$	$2.172 \times 10^{-1} \pm 3.00 \times 10^{-1}$
	SLNO	$6.758 \times 10^{-4} \pm 4.08 \times 10^{-6}$	$2.250 \times 10^{-3} \pm 7.96 \times 10^{-5}$	$2.320 \times 10^{-3} \pm 1.91 \times 10^{-4}$	$2.642 \times 10^{-3} \pm 1.44 \times 10^{-4}$
224×448	GSNO	$1.855 \times 10^{-4} \pm 1.62 \times 10^{-6}$	$9.467 \times 10^{-4} \pm 1.34 \times 10^{-4}$	$3.363 \times 10^{-3} \pm 9.69 \times 10^{-4}$	$3.637 \times 10^{-2} \pm 3.95 \times 10^{-2}$
	SFNO	$6.095 \times 10^{-4} \pm 9.11 \times 10^{-7}$	$2.255 \times 10^{-3} \pm 8.56 \times 10^{-5}$	$5.861 \times 10^{-3} \pm 3.36 \times 10^{-3}$	$1.700 \times 10^{-1} \pm 2.33 \times 10^{-1}$
	SLNO	$7.822 \times 10^{-4} \pm 2.82 \times 10^{-6}$	$2.253 \times 10^{-3} \pm 6.93 \times 10^{-5}$	$2.362 \times 10^{-3} \pm 1.88 \times 10^{-4}$	$2.667 \times 10^{-3} \pm 1.49 \times 10^{-4}$
256×512	GSNO	$9.934 \times 10^{-5} \pm 1.82 \times 10^{-6}$	$9.511 \times 10^{-4} \pm 1.47 \times 10^{-4}$	$3.480 \times 10^{-3} \pm 1.06 \times 10^{-3}$	$4.024 \times 10^{-2} \pm 4.40 \times 10^{-2}$
	SFNO	$5.972 \times 10^{-4} \pm 4.74 \times 10^{-7}$	$2.236 \times 10^{-3} \pm 8.53 \times 10^{-5}$	$5.869 \times 10^{-3} \pm 3.46 \times 10^{-3}$	$1.753 \times 10^{-1} \pm 2.40 \times 10^{-1}$
	SLNO	$6.422 \times 10^{-4} \pm 5.32 \times 10^{-6}$	$2.225 \times 10^{-3} \pm 7.44 \times 10^{-5}$	$2.349 \times 10^{-3} \pm 1.87 \times 10^{-4}$	$2.689 \times 10^{-3} \pm 1.30 \times 10^{-4}$

F.4 Zero-shot transfer conventions and complete results

Training is confined to 256×512. The target resolutions are 96×192, 128×256, 160×320, 192×384, 224×448, and 256×512. External training and evaluation grids are equiangular at every resolution. Internal Legendre–Gauss coordinates are part of the models' spectral implementation and do not change this external-grid contract. All learned weights and stored source-256 normalization statistics remain fixed at every target.

Model-specific transfer. On a new external grid, GSNO keeps the internal discretization of the training resolution, with dimensions 86×170 and maximum degree $\ell_{\max} = m_{\max} = 85$. SFNO retains the same internal discretization and the same fixed discrete-cosine basis in spherical degree. The learned weights of both models are used exactly as trained, without interpolation or remapping. SLNO changes only the deterministic spectral coordinates $\lambda_\ell = \ell(\ell+1)$ of the degrees retained at the target resolution. The learned map from these coordinates to the operator response is held fixed. The target resolution therefore enters solely by evaluating this map at the new coordinates, with no parameters refit to that resolution.

Target distributions and coverage. Initial conditions at each target resolution are drawn from that solver's native distribution of random initial data at Mach number 0.2. The spectral band admitted by this distribution changes with the grid, so the same sampling seed yields a different physical initial state at a different resolution. The comparison covers all $3 \times 6 \times 3 \times 3 = 162$ combinations of model, resolution, training seed, and evaluation seed. Resolutions other than the training grid were evaluated directly on the target grid; the native-resolution entries reuse earlier results whose agreement with this protocol had been verified. Each model–resolution pair comprises 450 trajectories, which are summarized by three training-seed means.

Range statistic. For model M, let $E_M(r)$ denote its 20 h mean error at target resolution r. We summarize the variation across the six evaluated target resolutions by

$$V_M = 100 \frac{\max_r E_M(r) - \min_r E_M(r)}{\min_r E_M(r)} \,\%. \tag{59}$$

Using the unrounded means gives $V_{\mathrm{SLNO}} = 1.7640\%$, $V_{\mathrm{GSNO}} = 22.8116\%$, and $V_{\mathrm{SFNO}} = 238.6853\%$. The corresponding SLNO maximum-to-minimum error ratio is 1.01764, showing that its 20 h error varies only modestly across the six evaluated target resolutions.

G Additional Global Weather Forecasting Details

G.1 Experimental protocol

GSNO and SLNO are trained on the same data, for the same duration, and under the same training protocol, and are evaluated using the same rollout procedure. The test period is 2017 and is divided into eight files. Each file provides 64 initial times, giving a total of 512 evaluation trajectories. Rollouts extend to a maximum lead time of 30 days.

The model input contains 27 atmospheric variables. Each forecast predicts six variables: 2 m temperature, 10 m zonal wind, 10 m meridional wind, 600 hPa zonal wind, 600 hPa meridional wind, and 600 hPa temperature. Each prediction advances the forecast by 24 hours.

The evaluation uses a partial-autoregressive protocol over these six predicted variables. At each rollout day, the corresponding 27-variable ERA5 state is normalized, the six predicted channels are replaced by the model outputs, and the remaining channels retain their observed ERA5 values. The resulting state is then used as input to the next forecast step. Consequently, prediction errors are recursively propagated through the six forecast variables, while the remaining atmospheric variables provide observed auxiliary inputs throughout the rollout.

Predictions and targets are compared in physical scale. Latitude weights proportional to $\cos(\text{latitude})$ are normalized to have mean one. The reported wRMSE aggregates errors over the six predicted variables under this latitude weighting. ACC is reported as a companion aggregate score from the same evaluation procedure.

The horizon-wise values in Table 3 are aggregate scores over the 512 partial-autoregressive evaluation trajectories.

G.2 Horizon-wise wRMSE and ACC

Table 3 reports wRMSE and ACC at eight forecast lead times. SLNO has lower wRMSE than GSNO at every reported horizon and higher ACC at every reported horizon.

At day 20, the relative wRMSE reduction is

$$\frac{3.459270 - 3.388283}{3.459270} \approx 2.05\%. \tag{60}$$

At day 30, the corresponding reduction is

$$\frac{3.455369 - 3.391095}{3.455369} \approx 1.86\%. \tag{61}$$

These results show that the improvement observed at short lead times is maintained throughout the evaluated 30-day partial-autoregressive rollout.

G.3 Qualitative sample protocol

The qualitative figures use a randomly selected trajectory from the 512-trajectory evaluation set. The trajectory was selected independently of the prediction errors of GSNO and SLNO. Both models are evaluated from the same initial state and under the same partial-autoregressive rollout protocol.

The displayed trajectory corresponds to initial time 338 in the eighth file of the 2017 test period, trajectory 505 of the evaluation. Figure 10 shows temperature at 600 hPa at lead times of 1, 10, 20, and 30 days. The example is used to visualize the spatial evolution of the forecasts and their errors; aggregate quantitative comparisons use all 512 evaluation trajectories.

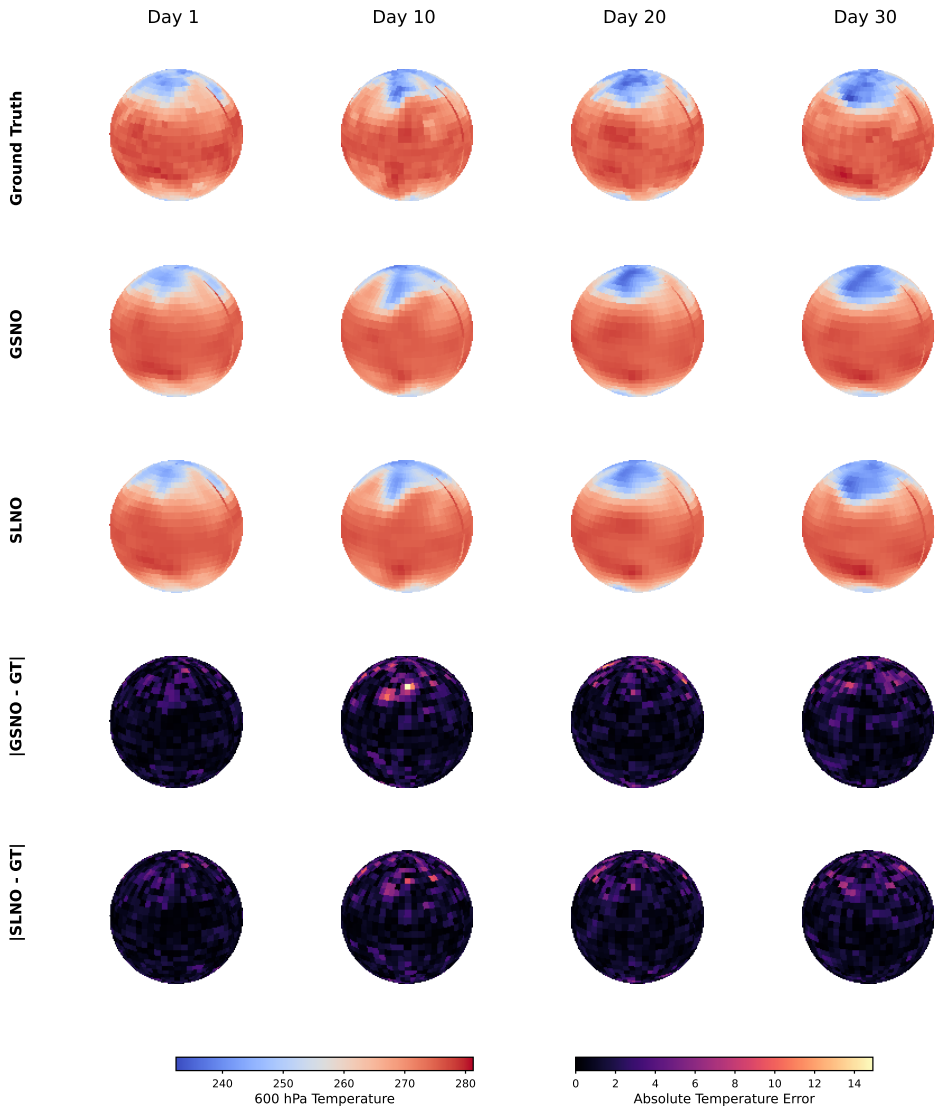

Figure 10: Partial-autoregressive rollout of temperature at 600 hPa on a randomly selected evaluation trajectory. The trajectory was selected independently of model prediction errors. The panels show forecasts at lead times of 1, 10, 20, and 30 days. Aggregate quantitative results over all 512 trajectories are reported in Table 3.

Table 11: wRMSE of temperature on the randomly selected qualitative trajectory.

Model	Day 1	Day 5	Day 10	Day 20	Day 30
GSNO	1.577938	2.053372	2.206836	2.227551	2.104265
SLNO	1.519735	1.941394	1.986965	1.867847	1.875959