WRAP: WIGNER ROTATION-EQUIVARIANT ATTENTION FOR PHYSICAL SIGNALS ON THE SPHERE

Anonymous authors
Paper under double-blind review

ABSTRACT

Many physical signals are spherical: a radar signature is a function of viewing direction, and an atmospheric state is a field on a rotating planet. Projecting them onto a plane distorts area, distance and orientation and discards the rotational symmetry of the underlying physical system. Previous equivariant attention methods either rely on computationally expensive steerable kernels and tensor products, limiting scalability, or, are restricted to Euclidean space, planar images or camera poses rather than spherical data. We introduce Wigner-RoPE, a spherical rotary positional encoding that generalizes rotary positional embeddings from translations along a line to rotations on the sphere. Wigner-RoPE represents lifted spherical positions as rotations from a reference direction using Wigner D-matrices. Building on Wigner-RoPE, we develop *W*igner *r*otation-equivariant *a*ttention for *p*hysical signals (WRAP), an expressive, multi-frequency, $SO(3)$-equivariant attention mechanism compatible with efficient fused attention kernels. Stacking WRAP layers yields WRAP-former, a transformer for spherical physical signals. Across radar classification, spherical segmentation, and medium-range weather forecasting, WRAP-former outperforms strong baselines with alternative positional embeddings and symmetry–efficiency tradeoffs. Finally, we compare spherical embedding methods with varying degrees of symmetry breaking, finding that different symmetry choices benefit different physical domains.

1 INTRODUCTION

Many scientific sensing signals have known physical structure: governing laws constrain their possible states and evolution. Radar observations and atmospheric fields, for example, are often natively defined on a sphere. Their natural transformation group is therefore the group of three-dimensional rotations, $SO(3)$, rather than the translations that commonly structure planar images and text. Treating spherical scientific signals as flat grids can obscure both their spherical geometry and their rotational symmetries. A transformer that treats the sphere as a flat grid of patches loses this structure twice: the projection distorts area, distance and orientation, and the rotational symmetry of the underlying physics has to be learned from data, usually by augmentation.

Architectures that explicitly encode symmetry incur costs elsewhere: spherical CNNs require spectral-domain convolution, and equivariant attention relies on tensor products instead of matrix multiplication. Neither fits the dense kernels optimized for transformers, so exact equivariance comes at the cost of the speed in every layer. Relative position encodings, by contrast, run on standard kernels, but existing versions only handle Euclidean translations or degree-one approximations on the sphere, not true rotations. Therefore, a key missing component is a spherical relative position encoding that works within the transformer's existing kernels.

We extend rotary position embeddings (RoPE) to the sphere. In the standard case, RoPE encodes position on the line as an offset, rotating queries and keys so attention depends only on their relative position(Su et al., 2024). The sphere has no global translation symmetry; its natural symmetry is rotation. Our method builds on this directly: each patch is assigned a rotation in $\mathrm{SO}(3)$, and queries and keys are transformed by the corresponding Wigner-D matrices, so every attention logit depends only on the relative rotation between two tokens. We also rotate values into the query's frame, which matters for vector-valued signals such as wind. Our method has the added benefit that it is highly efficient and scalable due to preserving flash-attention compatibility. Our contributions are:

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.

- We introduce a novel spherical positional encoding, Wigner-RoPE, that generalizes rotary positional embeddings from Euclidean translations to spherical rotations, enabling more expressive modeling of $SO(3)$ symmetry.
- We develop WRAP, an efficient spherical attention mechanism that incorporates this encoding without changing the core attention computation, making it compatible with existing transformer architectures and optimized attention kernels.
- We demonstrate the generality of the resulting WRAP-former architecture across diverse applications involving spherical physical signals, including radar classification, panoramic spherical segmentation, and weather forecasting, where our approach matches or outperforms strong baselines while offering favorable efficiency–symmetry tradeoffs.

2 Related Work

Positional encoding. RoPE (Su et al., 2024) rotates queries and keys so that attention logits depend on relative offsets on a line. Its extensions (Heo et al., 2024; Schenck et al., 2025) encode Euclidean coordinates, whose symmetry is translation. On the sphere, Unlu (2023) uses 3×3 rotations from latitude and longitude, a degree-one analogue without values or higher degrees, while SpheRoPE (Hirschorn et al., 2026) encodes coordinate differences rather than relative rotations. GTA (Miyato et al., 2024) and RoVE (García-Castellanos et al., 2026), like our method's value rotation component, transform values, but neither encodes exact relative rotations on the sphere.

Neural networks on the sphere. Spherical CNNs achieve rotation equivariance by convolving in the spectral domain (Cohen et al., 2018; Esteves et al., 2023), while gauge-equivariant CNNs operate in local frames (Cohen et al., 2019; Weiler et al., 2026). Spherical attention is approximately equivariant (Bonev et al., 2025b) or carries no guarantee (Carlsson et al., 2024; Benny & Wolf, 2025). Likewise no HEALPix-based weather transformer (Ramavajjala, 2024; Zhdanov et al., 2026) claims rotation equivariance. The closest, MOSAIC (Zhdanov et al., 2026), uses latitude–longitude RoPE; in contrast, our encoder represents rotations exactly.

Equivariant attention. Beyond approaches based on tensor products of steerable features (Fuchs et al., 2020; Liao & Smidt, 2023) or attention over group elements, several equivariant-attention architectures use standard kernels. (Hutchinson et al., 2021; Romero & Cordonnier, 2021), equivariant attention runs on standard kernels. GATr (Brehmer et al., 2023), octic ViTs (Nordström et al., 2025), Platonic Transformers (Islam et al., 2026) efficiently encode every frame of a point in the channels of a single token, while LLoCa (Spinner et al., 2025) transforms queries and keys into input- predicted frames. Closest to our method is the gauge equivariant transformer (He et al., 2021), which attends over regular-representation features of a cyclic group.

3 Background

Attention and position. For tokens $X \in \mathbb{R}^{N \times D}$ with queries $Q = XW_Q$, keys $K = XW_K$ and values $V = XW_V$, a self-attention head of dimension d_h computes $\mathrm{Attn}(Q, K, V) = \mathrm{softmax}(QK^\top/\sqrt{d_h})\,V$, which is permutation equivariant. Additive position embeddings (Vaswani et al., 2017; Dosovitskiy et al., 2021) break that symmetry by supplying positional information. RoPE (Su et al., 2024) is translation equivariant and rotates the query q_i and the key $k_i \in \mathbb{R}^{d_h}$ at position i, in every layer, by $\mathbf{B}_i = \mathrm{diag}(\mathbf{R}(i\,\omega_1), \ldots, \mathbf{R}(i\,\omega_{d_h/2}))$, with $\mathbf{R}(\phi)$ the 2×2 rotation by ϕ. Since $\mathbf{R}(\phi)^\top \mathbf{R}(\phi') = \mathbf{R}(\phi' - \phi)$,

$$\big(\mathbf{B}_i q_i\big)^\top \big(\mathbf{B}_j k_j\big) = q_i^\top \mathbf{B}_{j-i}\, k_j, \tag{1}$$

On the sphere, positions are not given by translations, so Section 4 replaces it with $SO(3)$ rotations (details in Appendix A).

Equivariance and lifting. A map F is equivariant for a group G acting on its inputs and outputs when $F(g \cdot f) = g \cdot F(f)$ for every $g \in G$. For signals on the sphere that group is $\mathrm{SO}(3)$, acting on $f : S^2 \to \mathbb{R}^C$ by $(g \cdot f)(y) = f(g^{-1}y)$. The sphere is not itself a group, but the homogeneous space $\mathrm{SO}(3)/\mathrm{SO}(2)$. Its points are the cosets of the rotations that fix the north pole $\hat{n} = (0, 0, 1)^\top$. As a result, two spherical points do not define a unique relative displacement that is itself a symmetry-group element; a relative positional encoding therefore requires a choice of lift or local frame.

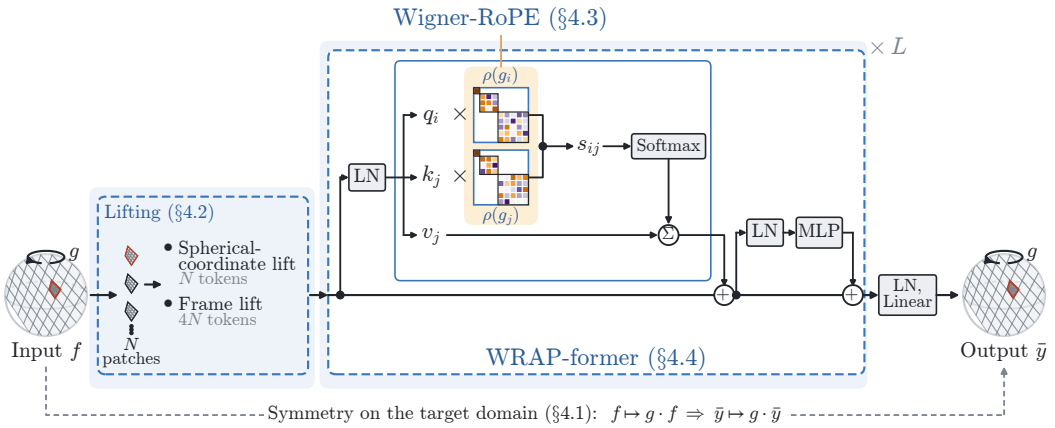


Figure 1: **WRAP-former architecture.** We lift patches of the spherical input f to tokens with $\mathrm{SO}(3)$ positions (§4.2) and process them with L WRAP-former blocks (§4.4), where the positions enter only through the Wigner-RoPE matrices $\rho(g)$ (orange, §4.3). Rotating the input by g rotates the output by the same g (§4.1).

Lifting removes the obstruction by carrying the signal from the sphere to the group (Cohen et al., 2018): a point y is represented by a rotation R with $R\hat{n} = y$, and two such rotations do have a relative rotation $R_i^{-1}R_j \in \mathrm{SO}(3)$. The representation is not unique, since R and $R\,R_z(\psi)$ carry $\hat{n}$ to the same point for every ψ, so choosing R means choosing a frame at y. No choice is continuous over the whole sphere, as a continuous global frame would trivialize the tangent bundle of S^2, and the frame of spherical coordinates is for instance undefined at the poles. A lift that assigns one frame per point is therefore equivariant only to a subgroup of $\mathrm{SO}(3)$, and keeping several frames per point recovers a larger portion of the full rotational symmetry (Section 4.2).

4 Method: WRAP-former

Our spherical transformer is WRAP-former, and it has three components. A *lift* turns a signal on the sphere into tokens, each carrying features and a rotation in $\mathrm{SO}(3)$ as its position (Figure 2); which lift we use decides which rotations the model is exactly equivariant to. *Wigner-RoPE* represents those rotations as orthogonal matrices, so position enters as a change of basis on queries and keys rather than as something the network has to learn. WRAP is Wigner-RoPE inside attention: every logit then depends on two tokens only through their relative rotation, and because each factor multiplies one token before the softmax, attention itself is unchanged and runs on the fused kernels an ordinary transformer already uses. We prove this section's results in Appendix E.

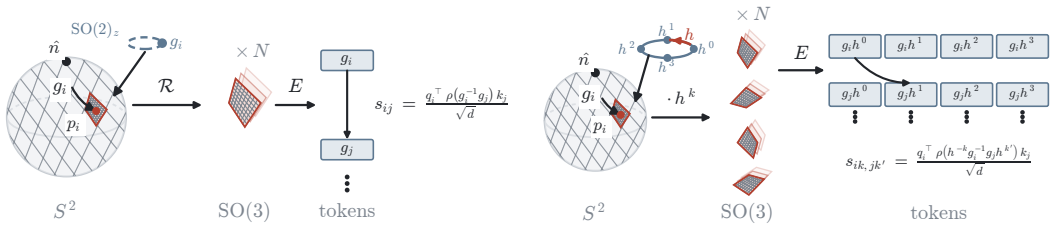
4.1 Symmetry on the target domain

We first can define our continuous setting with an encoder that we want to be equivariant to $SO(3)$ and then show the practical effects of discretization, because we only have access to the symmetries that map a patch onto itself, as a permutation. Let $G \subset \mathrm{SO}(3)$ be this group. Each $g \in G$ permutes the samples, acting on the sampled signal as $(g \cdot x)_b = x_{g^{-1}b}$. If Π_g denotes the corresponding permutation of encoder tokens, our equivariance requirement is

$$\mathcal{E}(g \cdot x) = \Pi_g \mathcal{E}(x) \qquad \text{for all } g \in G. \tag{2}$$

Thus, rotating the input moves each output token to the rotated location without changing its features. A permutation-invariant readout, such as the token mean, is consequently G-invariant.

The grid determines which rotations belong to G. For our HEALPix discretization, these are the four quarter turns about the polar axis, $G = C_4 = \{I, R_z(\pi/2), R_z(\pi), R_z(3\pi/2)\}$. We discuss the additional D_4 symmetry induced by the lift in Sections 4.2 and 4.5.



(a) Spherical-coordinate lift: one token per patch, in the frame of spherical coordinates.

(b) Frame lift: four tokens per patch, whose frames differ by quarter turns.

Figure 2: Two lifts from S^2 to $\mathrm{SO}(3)$. Every token carries a rotation, and the logit between two tokens depends on their rotations only through the relative rotation. The figure writes g_i for the rotation R_i of patch i and h for the quarter turn $R_z(\pi/2)$ about the patch center.

4.2 Lifting

We provide two lifts that are equivariant to a different class of symmetries with different cost profiles: which matters because matching the symmetry to the task is more important than simply choosing the largest possible group in real data with symmetry breaking, which Section 5 shows is the choice that matters.

We build a lift that turns the sampled signal into tokens, each with features from embedding the patch, and a position, a rotation that carries the north pole $\hat{n} = (0, 0, 1)^\top$ to the patch center and thereby fixes a frame there. The rotations that do this form a circle, since any one may be followed by a rotation about the center itself, so a center does not determine a frame and we must choose. In the limit of discretization, this would give an $\mathrm{SO}(3)$-equivariant lift, provided we express vector channels in the frame of each token. However, how we define the lift has direct effects on the cost profile, meaning depending on the problem domain different lifts have different tradeoffs, because real data often lacks exact symmetries in the physics (Appendix B.2). The spherical-coordinate lift keeps one frame per patch, that of spherical coordinates, and is exactly equivariant to the quarter turns about the polar axis, and the frame lift keeps four frames differing by quarter turns, and is exactly equivariant to the dihedral group D_4 they generate with the half turn (Section 4.5).

The spherical-coordinate lift. With colatitude and longitude (θ, ϕ), we give every point other than the poles the frame,

$$\mathcal{R}(\theta, \phi) \;=\; R_z(\phi)\, R_y(\theta) \;\in\; \mathrm{SO}(3), \tag{3}$$

where $\mathcal{R}(\theta, \phi)$ is the frame at the point with colatitude θ and longitude ϕ, and $R_z(\alpha)$ and $R_y(\alpha)$ are the rotations by α about the z and the y axis. We give patch i, whose center $\hat{n}_i$ has coordinates (θ_i, ϕ_i), the rotation $R_i = \mathcal{R}(\theta_i, \phi_i)$, so that $R_i \hat{n} = \hat{n}_i$ and the columns of R_i are the local south and east directions and the outward normal. Every patch is read with the same north. The relative rotation $R_i^{-1} R_j$ is the frame of patch j in the frame of patch i, and fixes both the direction from one to the other and their relative orientation (Appendix B.2).

Proposition 1 (The spherical-coordinate lift is $\mathrm{SO}(2)_z$-equivariant)**.** *For every $g \in \mathrm{SO}(2)_z$ and every $\hat{u} \in S^2$ other than the poles, $\mathcal{R}(g\hat{u}) = g\,\mathcal{R}(\hat{u})$. For a general $g \in \mathrm{SO}(3)$ with $g\hat{u}$ other than the poles, $\mathcal{R}(g\hat{u}) = g\,\mathcal{R}(\hat{u})\,h$ for some $h \in \mathrm{SO}(2)_z$ that depends on g and $\hat{u}$.*

Proposition 1 is the origin of the restriction to G: exactness requires that a rotation permute the pixels and that the lift be equivariant to it, and precisely the rotations in G do both.

The frame lift. The spherical-coordinate lift shows the encoder every patch in one frame, so it cannot be equivariant to a rotation that also turns the frames, such as the half turn $F = R_x(\pi)$ by π about the x axis, which exchanges the hemispheres and for which $F\,\mathcal{R}(\hat{u}) = \mathcal{R}(F\hat{u})\,R_z(\pi)$. The frame lift therefore represents every patch by four tokens whose frames differ by quarter turns about its center (Figure 2b). Token (i, k) carries,

$$R_{ik} \;=\; R_i\, R_z(k\pi/2), \qquad k \in \{0, 1, 2, 3\}, \tag{4}$$

and holds the values of patch i in that frame, with every tangent vector channel turned with it, so the spherical-coordinate lift is the case $k = 0$. Two of the four orientations are exact pixel permutations and two are a fixed conservative resampling, which commutes with G and F (Appendix C.1). The

frame lift therefore carries an exactness statement of its own: it is equivariant to the dihedral group D_4 generated by G and F, and invariant to the frame chosen at each patch (Proposition 3).

Sampling and cost. We sample the signal on a HEALPix grid (Górski et al., 2005), whose equal-area pixels the quarter turns about z permute with no interpolation, which is what makes (2) exact rather than approximate (Appendix B.1). The frame lift pays for its extra frames in tokens: against the spherical-coordinate lift it costs $4\times$ in linear layers and $16\times$ in attention, and Appendix C.2 measures what that is worth per training step.

4.3 WIGNER-ROPE

Wigner-RoPE is what allows us to build an equivariant model by generalizing RoPE to the sphere and rotations as position, while maintaining compatibility with flash-attention style kernels. Our positional encoding represents each rotation R_i by a matrix $\rho(R_i)$ that acts inside an attention head. We require ρ to be a group homomorphism, so that composing two positions composes their matrices, with $\rho(R)^\top = \rho(R^{-1})$, so that every $\rho(R)$ is orthogonal and preserves the norms of queries and keys. Real Wigner-D matrices satisfy both: for $R \in \mathrm{SO}(3)$ and a degree $\ell \geq 0$, $D^\ell(R) \in \mathbb{R}^{(2\ell+1)\times(2\ell+1)}$ is the orthogonal matrix by which R acts on the real spherical harmonics of degree ℓ (Appendix D), and $D^0(R) = 1$. We fill every head with Wigner-D blocks of degrees 1 to L and $n_0 \geq 1$ trivial channels,

$$\rho(R) \;=\; I_{n_0} \oplus \bigoplus_{\ell=1}^{L} D^\ell(R), \qquad n_0 + (L+1)^2 - 1 = d_h, \tag{5}$$

where d_h is the head dimension, L is the largest degree the head holds, I_{n_0} is the $n_0 \times n_0$ identity, i.e., n_0 copies of D^0, and $\oplus$ stacks blocks along the diagonal. The trivial channels let d_h take any value, and all our models use $n_0 = 1$, with d_h and L per model in Appendix C.2. By construction ρ is an orthogonal representation, so $\rho(R)^\top \rho(R) = I$. Rotations also leave the trivial channels unchanged. More precisely, writing $P = I_{n_0} \oplus 0$ for the orthogonal projection onto them, we have,

$$\rho(R_1 R_2) = \rho(R_1)\rho(R_2), \qquad \rho(R)^\top = \rho(R^{-1}), \tag{6}$$

$$P\,\rho(R) = \rho(R)\,P = P \qquad \text{for all } R, R_1, R_2 \in \mathrm{SO}(3). \tag{7}$$

4.4 WRAP: EQUIVARIANT WIGNER-ROTARY ATTENTION

We formally integrate Wigner-RoPE into attention, which is WRAP, which recovers an efficient flash-attention compatible expressive provably $SO(3)$ equivariant attention mechanism.

Rotary attention. Let $q_i, k_i, v_i \in \mathbb{R}^{d_h}$ be the query, key and value of token i in one head, formed by linear maps shared by all tokens, and let R_i be its rotation, i.e., R_{ik} for token (i,k) of the frame lift, where the subscript k of R_{ik} always denotes an orientation and k_j always denotes a key. We rotate queries and keys by ρ of their rotations and apply standard attention, so by (6) the logits s_{ij} and head outputs out_i are,

$$\tilde{q}_i = \rho(R_i)\,q_i, \qquad \tilde{k}_j = \rho(R_j)\,k_j, \tag{8}$$

$$s_{ij} = \tilde{q}_i^\top \tilde{k}_j / \sqrt{d_h} = q_i^\top\, \rho\big(R_i^{-1} R_j\big)\, k_j / \sqrt{d_h}, \tag{9}$$

$$\mathrm{out}_i = \textstyle\sum_j \mathrm{softmax}_j(s_{ij})\, v_j. \tag{10}$$

where $\tilde{q}_i$ and $\tilde{k}_j$ are the rotated query and key, s_{ij} is the logit between tokens i and j, out_i is the output of the head at token i, and $\mathrm{softmax}_j$ normalizes over the key index j. That is, the logit depends on R_i and R_j only through their relative rotation, never on either absolute position. WRAP is the spherical analogue of RoPE, with $\mathrm{SO}(2)$ and its 2×2 blocks replaced by $\mathrm{SO}(3)$ and its Wigner-D matrices, and the offset $j - i$ of (1) by $R_i^{-1} R_j$.

Value rotation. A head adds values expressed in different frames, in which a direction has different components, so we rotate each value into the frame of its query, which by (6) gives,

$$\mathrm{out}_i = \rho(R_i)^\top \textstyle\sum_j \mathrm{softmax}_j(s_{ij})\, \rho(R_j)\, v_j = \sum_j \mathrm{softmax}_j(s_{ij})\, \rho\big(R_i^{-1} R_j\big)\, v_j. \tag{11}$$

Both rotations act outside the softmax, so compatibility with optimized fused kernel like flash-attention is preserved. Value rotation is effectively a null operation on a scalar, matters for a direction such as the wind, and leaves the symmetry of the model unchanged (Propositions 2 and 8). It also

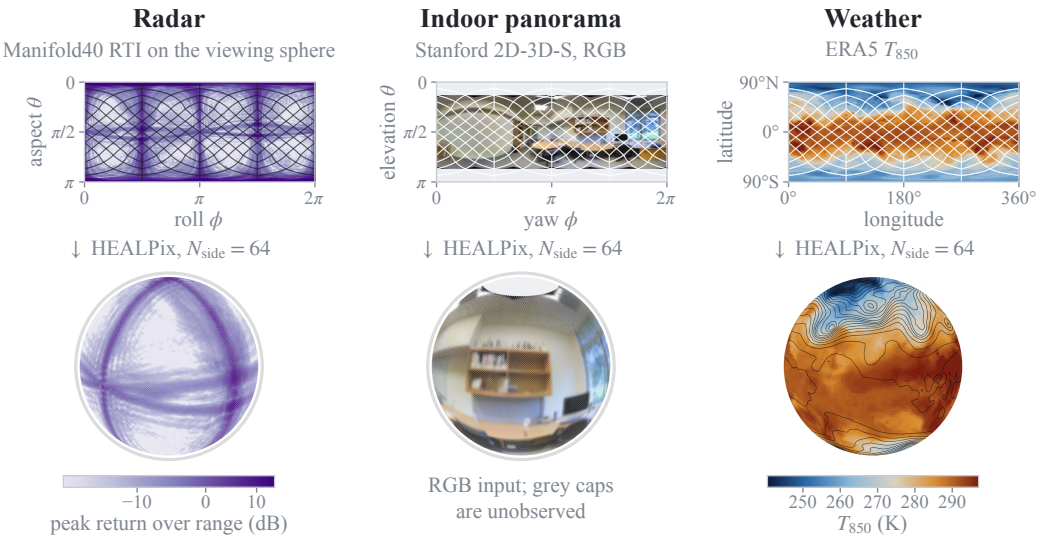


Figure 3: The three spherical domains, each shown on its native grid (top) and resampled onto HEALPix at `nside` $= 64$ (bottom).

matters more for frame lift, because multiple tokens per patch means those frames need to be rotated into each other, or vectors will lose tangential information. We test and verify the importance of this in Appendix H.7 on weather-bench.

Proposition 2 (Rotary attention is equivariant)**.** *Let A_R be a rotary attention layer, with or without value rotation, whose tokens carry the rotations $R = (R_i)_i$, and let Π be the matrix of a permutation π of the tokens. If $R'_{\pi(i)} = SR_i$ for every token i and one $S \in \mathrm{SO}(3)$, then $A_{R'}(\Pi Z) = \Pi\, A_R(Z)$ for every sequence Z of token features.*

For $g \in G$, the grid fixes the rotations, i.e., $R' = R$, and Proposition 1 gives $R_{\pi_g(i)} = gR_i$, i.e., $S = g$, so rotary attention is G-equivariant. Our encoder stacks blocks of rotary attention and a pointwise MLP, with normalization and residual connections, which act on each token separately and commute with every Π, so every block is G-equivariant, also with a key and value fixed at zero as in our weather models, since $\rho(R)\,0 = 0$ (Appendix H.1). We give a CLS token, which carries no position, the projector P of (7) in place of a rotation, and its embedding is G-invariant (Proposition 5 in Appendix B.3).

Theorem 1 (The encoder is G-equivariant)**.** *For either lift and every $g \in G$, $\mathcal{E}(g \cdot x) = \Pi_g\, \mathcal{E}(x)$, with and without value rotation and with or without a* CLS *token, which Π_g leaves in place.*

4.5 Output, loss and symmetry of the frame lift

For frame lift, we define the output in the case of multiple orientations per patch, and how the readout should be structured, which differs from training, and the resulting symmetry for frame lift. The final layer predicts every patch once per orientation. We take the output $\bar{y}$ to be the mean of the predictions of orientations 0 and 2, the two that lie at the pixel centers, with orientation 2 mapped back to the frame of orientation 0, and a rollout takes $\bar{y}$ as its next input (Appendix C.3). We train each orientation against the target in its own frame, because loss on the readout may allow errors to cancel in the mean. (Appendix C.4). The models on HEALPix are approximately equivariant to $SO(3)$ rotations outside D_4, and north-aligned frames of (3) are an $SO(3)$ symmetry breaking prior away from the polar axis, which we compare to a block-frames variant that does not.

Proposition 3 (Symmetry of the frame lift)**.** *The frame lift is exactly equivariant to the dihedral group $D_4 = \langle R_z(\pi/2), F\rangle$ of order eight, generated by G and the half turn F, and no subgroup of $\mathrm{SO}(3)$ larger than D_4 permutes the HEALPix pixels. Its output is invariant to the choice of frame at every patch, i.e., to the gauge group C_4^N of changes of frame $R_i \mapsto R_i\, R_z(\pi/2)$ at any patch, which cyclically relabel the four tokens of the patch together with their pixel positions. The spherical-coordinate lift is in general not equivariant to F.*

Table 1: **Manifold40 radar classification.** Top-1 / Top-5 accuracy (%) from a linear probe on the frozen encoder.

Model	Top-1 (↑)	Top-5 (↑)
ViT	83.41 ± 0.99	97.82 ± 0.10
Attn. on Sphere	80.32 ± 0.35	96.24 ± 0.18
SFNO	80.50 ± 0.66	96.15 ± 0.08
WRAP-former SC-lift	84.43 ± 0.34	97.82 ± 0.05
WRAP-former Frame lift	$\mathbf{85.40 \pm 1.17}$	$\mathbf{98.03 \pm 0.32}$

Table 2: **Stanford 2D3DS segmentation.** Mean IoU and accuracy (%) on the lat-lon grid.

Model	mIoU (↑)	Acc. (↑)
Flat transformer		
S2 Transformer	0.623 ± 0.161	96.5 ± 2.0
WRAP-former SC-lift	$\mathbf{0.701 \pm 0.169}$	$\mathbf{97.30 \pm 1.92}$
WRAP-former Frame lift	0.692 ± 0.169	97.20 ± 1.90
Hierarchical SegFormer		
S2 SegFormer	0.645 ± 0.132	96.8 ± 1.5
WRAP-former SC-lift	0.681 ± 0.175	97.08 ± 2.01
WRAP-former Frame lift	0.684 ± 0.173	97.13 ± 1.98

5 Experiments

5.1 Radar Classification

Task. The objective of the radar benchmark is to classify an object from its spherical radar signature. We train in two stages. To learn radar representations without relying on labels, we first pretrain the encoder with the masked autoencoder objective of reconstructing the masked signal. We then freeze the encoder and train a linear classifier on its mean-pooled patch token features.

Dataset. We build our radar dataset from the meshes of Manifold40 (Hu et al., 2022), a 40-class benchmark of 3D shapes. For every mesh we simulate a high-frequency, far-field radar signature with a physical optics simulator (Goodwin, 2026; Balanis, 2012). The signature is spherical because it is simulated over both body axes of the mesh at 180 orientations per axis, and each viewing direction carries a range profile of 83 bins, so a sample is a 83-channel signal on the sphere. We provide the training details in Appendix F.1

Results. We compare against three baselines: a standard ViT with the same transformer backbone and a learned positional encoding, the spherical transformer of Bonev et al. (2025b), and SFNO (Bonev et al., 2023) trained from scratch. Table 1 reports linear-probe accuracy on the held-out test set. WRAP-former with the frame lift outperforms all baselines in both Top-1 and Top-5 classification accuracy. We also see a larger gain over the baselines from the frame lift than from the spherical-coordinate lift, indicating that the additional orientations of the frame lift help learn better features for the downstream classification task. Additionally, we report the reconstruction loss and qualitative results for radar pretraining in Appendix F.2.

5.2 Spherical Image Segmentation

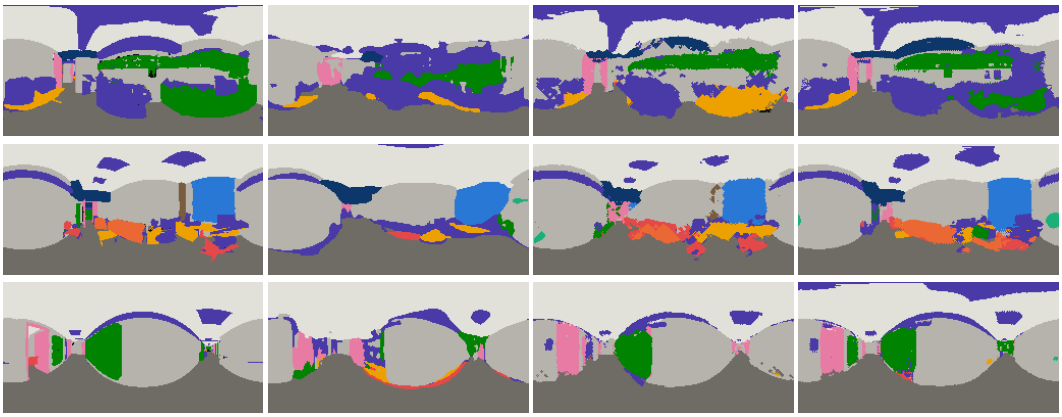
Task. The objective of the spherical segmentation benchmark is to assign a semantic class to every direction on the sphere. The benchmark is difficuilt

Dataset. We use the Stanford 2D3DS dataset (Armeni et al., 2017), 1413 indoor equirectangular panoramas with a dense semantic label per pixel over 13 classes. The panoramas are downsampled to a 128×256 equiangular grid. We resample each panorama from lat-lon onto a HEALPix grid (Górski et al., 2005) at $\texttt{nside} = 64$ (49152 equal-area pixels), bilinearly for the RGB channels and by nearest neighbour for the labels, so that class identities and the ignore index survive the transfer intact. We provide the training details in Appendix G.

Results. We compare against the spherical segmentation models of Bonev et al. (2025b). Following their setup, we evaluate our model in two variants, a flat transformer and a hierarchical SegFormer. Both variants outperform the corresponding baseline in mean IoU and accuracy, and the strongest result overall is the spherical-coordinate lift with the flat transformer as shown in table 2. We also provide the qualitative results in Fig. 4.

5.3 Weather Dynamics Regression

Task We evaluate WRAP-former on the WeatherBench 2 framework for medium-range atmospheric forecasting up to 14 days out. A key limitation of our current $\mathrm{SO}(3)$-equivariant architecture



(a) Ground truth (b) S2 SegFormer (c) WRAP-Former frame lift (d) WRAP-Former SC lift

Figure 4: **Stanford 2D3DS segmentation.** Predictions on three held-out test scenes, compared against the S2 SegFormer of Bonev et al. (2025b).

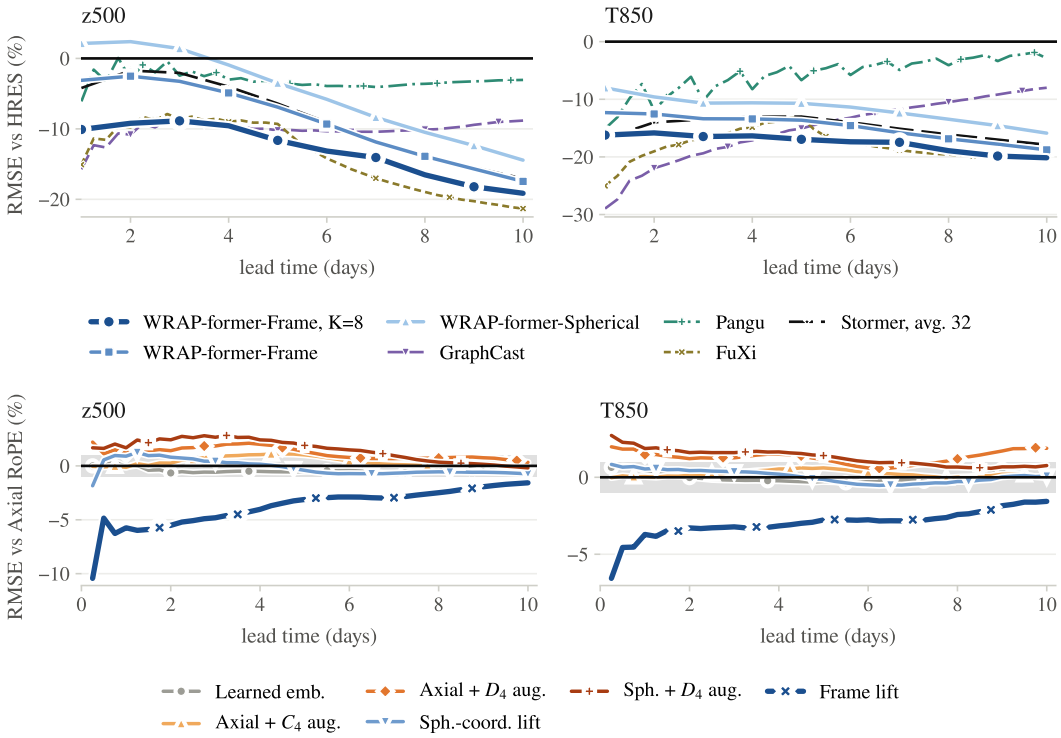


Figure 5: Top: RMSE change against IFS HRES on the zero line, 2020, for WRAP-former-Frame at $K = 8$ and as a single rollout, WRAP-former-Spherical, and four published systems, where negative is better. All curves are single rollouts except two step-sequence averages: Stormer's best 32 of 128 chosen on validation, ours 8 randomly chosen. Bottom: the controlled comparison as 6-hour rollouts, against axial RoPE, with the $\pm 1\%$ run-to-run band. The frame lift is drawn heavier.

is that it assumes ideal spherical invariance, neglecting real-world symmetry-breaking dynamics. In Earth's atmosphere, the Coriolis effect privileges the polar axis, breaking global symmetry into distinct latitudinal bands. Additionally, there are correct symmetries outside $SO(3)$ that our method doesn't natively consider, such as equatorial reflections ($\mathbb{Z}_2$), which dictate hemispheric dynamics. (Appendix H.1).

Dataset and training. ERA5 (Hersbach et al., 2020) for 1979–2018 at the 1.5° resolution of WeatherBench 2 (Rasp et al., 2024), resampled to HEALPix with $N_{\text{side}} = 64$. We select checkpoints on 2019 and score 2020 under the WeatherBench 2 protocol. Below we report geopotential at 500 hPa (z500), temperature at 850 hPa (T850) and 2 m temperature (T2m). Appendix H.9 carries other results. All models build on but make significant changes to Stormer (Nguyen et al., 2024), differing within each comparison only in encoding, lift, value rotation and augmentation. A lift name means the first stage alone, WRAP-former that stage plus rollout training (Appendix H).

Results. WRAP-former-Frame is competitive with published systems while training at a lower resolution. Its 5-day z500 RMSE is $284\,\mathrm{m}^2\,\mathrm{s}^{-2}$, -6.9% against IFS HRES, where GraphCast and Pangu-Weather train at 0.25° and we train at 1.5° (Figure 5, top, and Table 8). Averaging $K = 8$ step sequences, we beat Stormer on z500 at every lead by 2.5 to 6.9% and on T850 from 3 days. T2m is level from 5 days, and Stormer leads it at shorter range, even though Stormer averages the best 32 of 128 sequences chosen on validation, while our 8 are fixed in advance (Appendix H.4). The forecasts are also physically well behaved. At 5 days they hold 0.93–0.97 of ERA5's power at planetary and synoptic scales, though only 0.47 at the mesoscale, and their hydrostatic and continuity residuals sit at or below ERA5's own, with only continuity rising above it by 10 days (Appendix H.12). Perturbing the initial state gives a calibrated $M = 50$ ensemble, which Appendix H.5 reports.

Controlled comparison. Making a correct symmetry exact for the model beats data augmentation in our setting and incorrect symmetry hurts due to symmetry breaking. Full details are given in but protocol is in 6. Rotation augmentation raises 5-day error in every model we apply it to: by 1.0–2.3% for axial RoPE and by 2.3% for the spherical-coordinate lift under D_4. Full $\mathrm{SO}(3)$ augmentation destabilizes training and degrades performance. Making the group exact instead puts us 1.4% (C_4) and 4.4% (D_4) ahead of axial RoPE augmented with the same group. The frame lift lowers 5-day z500 error by 2.7% against the spherical-coordinate lift and by 2.6% against learned embeddings, the best non equivariant baseline. Equatorial data augmentation gains 0.7–1.4% on every variable and lead as a correct symmetry, while removing value rotation raises error throughout on frame lift, by about a percent for 850 hPa wind, a vector field that requires value rotation to match frames. Symmetry breaking matters even where the symmetry isn't grid-exact: the block-frames variant lets each pixel block define "up", losing north pole alignment's approximate $SO(3)$ symmetry breaking on the grid and raising 5-day z500 error by 3.2% on 2019, because north pole alignment aligns with the Coriolis effect. Full results are in Figure 5 and Appendices H.6, H.7.

6 Conclusion

In this work, we present WRAP-former, a spherical transformer built on Wigner-RoPE, an $\mathrm{SO}(3)$ positional encoding and two lifts that carry a signal from S^2 to $\mathrm{SO}(3)$: a spherical-coordinate lift and a frame lift. The resulting architecture is equivariant to $SO(3)$ rotations approximately and D_4 rotations exactly on HEALPix grids while retaining compatibility with efficient transformer implementations. We evaluate it on three domains, radar classification, spherical segmentation and weather forecasting. On radar benchmark, a linear probe classifier finetuned on our pretrained encoder improves classification accuracy over other baselines. For spherical segmentation, our WRAP-former improves mean IoU over the corresponding baseline in both the transformer and the hierarchical SegFormer, indicating the gain comes from the positional encoding rather than from a particular architecture. On WeatherBench 2, WRAP-former-Frame is competitive with published high-resolution forecasting systems despite training at 1.5° and outperforms HRES. Controlled comparisons show that building in the right symmetry beats augmentation, while imposing the wrong symmetry or ignoring physical symmetry breaking worsens forecasts. These results suggest that efficient, symmetry-aware positional encodings offer a practical path for transformers operating on diverse spherical physical signals.

Although our method is designed to be $SO(3)$ equivariant, in practice it incurs some equivariance error due to the discretization of the spherical grid. In future work, we plan to reduce this error by considering spherical grids other than HEALPix, which could be lifted to larger subgroups of $SO(3)$ than D_4. We also consider only equivariance to $SO(3)$ here, but our theory for RoPE and attention may extended to other orthogonal and non-orthogonal groups for non-spherical input signals.

AI USE STATEMENT

In this work, we used generative AI tools for polishing text, producing figures, and writing code. We have not used generative AI tools to generate the core research idea, the method, or the theoretical results of this work, and the remaining disclosure categories are not applicable to this work. We have reviewed all AI-assisted work. All AI-assisted text was read and edited by the authors, figures were checked against the underlying data, and AI-written code was verified and tested by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study involves no human participants, animal subjects, or private user data, and relies entirely on standard public research benchmarks. The proposed geometric learning methodology conforms strictly to institutional research ethics guidelines.

REPRODUCIBILITY STATEMENT

The method is described in full in Section 4, and the hyperparameters for every experiment are reported in the main text and in the Appendix. Dataset construction and preprocessing are described in Section 5. We plan to release the source code publicly upon acceptance. This will ensure that anyone will be able to reproduce our work.

REFERENCES

Iro Armeni, Sasha Sax, Amir R. Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

Constantine A Balanis. *Advanced engineering electromagnetics*. John Wiley & Sons, 2012.

Yaniv Benny and Lior Wolf. SphereUFormer: A u-shaped transformer for spherical 360 perception. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 940–950, 2025. URL https://arxiv.org/abs/2412.06968.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619:533–538, 2023. URL https://arxiv.org/abs/2211.02556.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, Jayesh K. Gupta, Kit Thambiratnam, Alexander T. Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. A foundation model for the Earth system. *Nature*, 641(8065):1180–1187, 2025. doi: 10.1038/s41586-025-09005-y.

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 (ICML)*, 2023. URL https://arxiv.org/abs/2306.03838.

Boris Bonev, Thorsten Kurth, Ankur Mahesh, et al. FourCastNet 3: A geometric approach to probabilistic machine-learning weather forecasting at scale. *arXiv preprint arXiv:2507.12144*, 2025a.

Boris Bonev, Max Rietmann, Andrea Paris, Alberto Carpentieri, and Thorsten Kurth. Attention on the sphere. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. URL https://arxiv.org/abs/2505.11157.

Johann Brehmer, Pim de Haan, Sönke Behrends, and Taco Cohen. Geometric algebra transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. URL https://arxiv.org/abs/2305.18415.

Oscar Carlsson, Jan E. Gerken, Hampus Linander, Heiner Spieß, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. HEAL-SWIN: A vision transformer on the sphere. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6067–6077, 2024. doi: 10.1109/CVPR52733.2024.00580. URL https://arxiv.org/abs/2307.07313.

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

Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018.

Taco S. Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1902.04615.

Guillaume Couairon, Renu Singh, Anastase Charantonis, Christian Lessig, and Claire Monteleoni. ArchesWeather & ArchesWeatherGen: a deterministic and generative model for efficient ML weather forecasting. *arXiv preprint arXiv:2412.12971*, 2024.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Carlos Esteves, Jean-Jacques Slotine, and Ameesh Makadia. Scaling spherical CNNs. In *International Conference on Machine Learning (ICML)*, 2023.

Christopher A. T. Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014. doi: 10.1002/qj.2270.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Alejandro García-Castellanos, Maurice Weiler, and Erik J. Bekkers. RoVE: Rotary value embeddings attention for relative position-dependent value pathways. In *Topology, Algebra, and Geometry in Data Science*, 2026. URL https://openreview.net/forum?id=0148YNqoZ6.

Justin Goodwin. Rfsim, 2026. URL https://github.com/mit-ll-responsible-ai/rfsim_mesh.

Krzysztof M. Górski, Eric Hivon, Anthony J. Banday, Benjamin D. Wandelt, Frode K. Hansen, Martin Reinecke, and Matthias Bartelmann. HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, 622(2): 759–771, 2005.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision (ECCV)*, 2024.

Hans Hersbach, Bill Bell, Paul Berrisford, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

Or Hirschorn, Aaron Olender, Eli Alshan, Ianir Ideses, Lior Fritz, and Sagie Benaim. SpheRoPE: Zero-shot optimization-free 360 panorama generation with spherical RoPE. *arXiv preprint arXiv:2606.32033*, 2026. URL https://arxiv.org/abs/2606.32033.

Shi-Min Hu, Zheng-Ning Liu, Meng-Hao Guo, Junxiong Cai, Jiahui Huang, Tai-Jiang Mu, and Ralph R. Martin. Subdivision-based mesh convolution networks. *ACM Trans. Graph.*, 41(3): 25:1–25:16, 2022. URL https://doi.org/10.1145/3506694.

Yifan Hu, Fukang Yin, Weimin Zhang, Kaijun Ren, Junqiang Song, Kefeng Deng, and Di Zhang. Spherical multigrid neural operator for improving autoregressive global weather forecasting. *Scientific Reports*, 15:11522, 2025. doi: 10.1038/s41598-025-96208-y. URL https://www.nature.com/articles/s41598-025-96208-y.

Michael J. Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. LieTransformer: Equivariant self-attention for Lie groups. In *International Conference on Machine Learning (ICML)*, 2021.

Mohammad Mohaiminul Islam, Rishabh Anand, David R. Wessels, Friso de Kruiff, Thijs P. Kuipers, Rex Ying, Clara I. Sánchez, Sharvaree Vadgama, Georg Bökman, and Erik J. Bekkers. Platonic transformers: A solid choice for equivariance. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/64816.

Alan Jeffares, Tennison Liu, Jonathan Crabbé, and Mihaela van der Schaar. Joint training of deep ensembles fails due to learner collusion. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Matthias Karlbauer, Nathaniel Cresswell-Clay, Dale R. Durran, Raul A. Moreno, Thorsten Kurth, Boris Bonev, Noah Brenowitz, and Martin V. Butz. Advancing parsimonious deep learning weather prediction using the HEALPix mesh. *Journal of Advances in Modeling Earth Systems*, 16, 2024.

Ryan Keisler. Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*, 2022.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, et al. Neural general circulation models for weather and climate. *Nature*, 632:1060–1066, 2024.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. URL https://arxiv.org/abs/2212.12794.

Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana C. A. Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, Zied Ben Bouallègue, Ana Prieto Nemesio, Peter D. Dueben, Andrew Brown, Florian Pappenberger, and Florence Rabier. AIFS – ECMWF's data-driven forecasting system. *arXiv preprint arXiv:2406.01465*, 2024. URL https://arxiv.org/abs/2406.01465.

Martin Leutbecher. Ensemble size: How suboptimal is less than infinity? *Quarterly Journal of the Royal Meteorological Society*, 145:107–128, 2019. doi: 10.1002/qj.3387.

Martin Leutbecher and Tim N. Palmer. Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539, 2008. doi: 10.1016/j.jcp.2007.02.014.

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3D atomistic graphs. In *International Conference on Learning Representations (ICLR)*, 2023.

Hampus Linander, Tage Tykesson, Pietro Rosso, Christoffer Petersson, Daniel Persson, and Jan E. Gerken. PEAR: Equal area weather forecasting on the sphere. *arXiv preprint arXiv:2505.17720*, 2025. URL https://arxiv.org/abs/2505.17720.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations (ICLR)*, 2024.

Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcomano, Sandeep Madireddy, Romit Maulik, Veerabhadra Kotamarthi, Ian Foster, and Aditya Grover. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 68740–68771, 2024.

David Nordström, Johan Edstedt, Fredrik Kahl, and Georg Bökman. Quick ViTs: Speeding up vision transformers through equivariance. *arXiv preprint arXiv:2505.15441*, 2025. URL https://arxiv.org/abs/2505.15441.

Isidoro Orlanski. A rational subdivision of scales for atmospheric processes. *Bulletin of the American Meteorological Society*, 56(5):527–530, 1975. doi: 10.1175/1520-0477-56.5.527.

Joel Oskarsson, Tomas Landelius, Marc Peter Deisenroth, and Fredrik Lindsten. Probabilistic weather forecasting with hierarchical graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022. URL https://arxiv.org/abs/2202.11214.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044): 84–90, 2025. doi: 10.1038/s41586-024-08252-9.

Vivek Ramavajjala. HEAL-ViT: Vision transformers on a spherical mesh for medium-range weather forecasting. *arXiv preprint arXiv:2403.17016*, 2024.

Stephan Rasp, Stephan Hoyer, Alexander Merose, et al. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024.

Mark J. Rodwell, David S. Richardson, Tim D. Hewson, and Thomas Haiden. A new equitable score suitable for verifying precipitation in numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 136(650):1344–1363, 2010. doi: 10.1002/qj.656.

David W. Romero and Jean-Baptiste Cordonnier. Group equivariant stand-alone self-attention for vision. In *International Conference on Learning Representations (ICLR)*, 2021.

Salva Rühling Cachay, Brian Henn, Oliver Watt-Meyer, Christopher S. Bretherton, and Rose Yu. Probabilistic emulation of a global climate model with spherical DYffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.14798.

Salva Rühling Cachay, Duncan Watson-Parris, and Rose Yu. U-Cast: A surprisingly simple and efficient frontier probabilistic AI weather forecaster. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2604.09041.

Connor Schenck, Isaac Reid, Mithun George Jacob, Alex Bewley, Joshua Ainslie, et al. Learning the RoPEs: Better 2D and 3D position encodings with STRING. In *International Conference on Machine Learning (ICML)*, 2025.

Jonas Spinner, Luigi Favaro, Peter Lippmann, Sebastian Pitz, Gerrit Gerhartz, Tilman Plehn, and Fred A. Hamprecht. Lorentz local canonicalization: How to make any network Lorentz-equivariant. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.20280.

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

Christopher Subich, Syed Zahid Husain, Leo Separovic, and Jing Yang. Fixing the double penalty in data-driven weather forecasting through a modified spherical harmonic loss function. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57191–57211, 2025.

Eren Unlu. Spherical position encoding for transformers. *arXiv preprint arXiv:2310.04454*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. *Equivariant and Coordinate Independent Convolutional Networks: A Gauge Field Theory of Neural Networks*, volume 1 of *Progress in Data Science*. World Scientific, 2026. ISBN 9789819806621. doi: 10.1142/14143.

WindBorne Systems. WeatherMesh-4 technical blog. `https://windbornesystems.com/blog/weathermesh-4-technical-blog`, 2025.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, et al. Small-scale proxies for large-scale transformer training instabilities. In *International Conference on Learning Representations (ICLR)*, 2024.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Maksim Zhdanov, Ana Lucic, Max Welling, and Jan-Willem van de Meent. (sparse) attention to the details: Preserving spectral fidelity in ml-based weather forecasting models. In *International Conference on Machine Learning (ICML)*, 2026. URL `https://arxiv.org/abs/2604.16429`.

A Position encoding in attention

Section 3 states the relative property of RoPE (Su et al., 2024). This appendix adds the frequencies it omits. RoPE rotates each consecutive pair of coordinates of a query or a key by an angle proportional to the position. With d_h even, $\omega_c = b^{-2(c-1)/d_h}$ for $c = 1, \dots, d_h/2$ and base b, typically $b = 10^4$, we write that matrix for token i as,

$$\mathbf{B}_i = \operatorname{diag}\big(\mathbf{R}(i\,\omega_1), \dots, \mathbf{R}(i\,\omega_{d_h/2})\big) \in \mathbb{R}^{d_h \times d_h}, \tag{12}$$

so consecutive pairs turn at geometrically spaced rates. Additive position embeddings (Vaswani et al., 2017; Dosovitskiy et al., 2021) instead encode absolute positions, so a model can depend on where a token is and not only on where it is relative to the others. RoPE leaves the values unrotated, which value rotation (11) changes.

B Details of the encoder

B.1 Spherical sampling and patches

We use HEALPix in preference to an equirectangular grid for two reasons. First, its pixels are equal-area, so no region of the sphere is oversampled. Second, G is a subgroup of its symmetry group: a quarter turn about z acts on HEALPix pixel indices by a permutation and requires no interpolation, which is what makes the guarantee in (2) exact rather than approximate. When the signal is supplied on a regular (θ, ϕ) grid, we resample it onto the HEALPix pixel centers by bilinear interpolation. Provided the longitude axis is divided into a multiple of four columns, a quarter turn acts on that grid as a rigid shift of columns, so the resampling commutes with G exactly and (2) is unaffected. For a general rotation, the resampling is only approximately equivariant.

In NESTED ordering, the pixel index is a quadtree address, so that for a patch resolution M_{side} that divides N_{side},

$$\text{pixel} \;=\; i \cdot p_p + u, \qquad N = 12\,M_{\text{side}}^2, \qquad p_p = \left(N_{\text{side}}/M_{\text{side}}\right)^2, \tag{13}$$

where N_{side} sets the resolution of the grid, which carries $N_{\text{pix}} = 12\,N_{\text{side}}^2$ pixels, M_{side} sets the resolution of the patch grid, N is the number of patches, p_p is the number of pixels per patch, C is the number of channels, i indexes the patch and u indexes the pixel within it. Patchification is therefore a pure reshape, with no gather and no interpolation, and a patch is both a contiguous block of pixel indices and a contiguous region of the sphere. The features of a token are a linear embedding of the p_pC values of its patch, with one map shared by all patches, $\mathbb{R}^{C\times N_{\text{pix}}} \to \mathbb{R}^{N\times p_pC} \to \mathbb{R}^{N\times D}$, where D is the encoder width. This embedding is G-equivariant. A quarter turn about z permutes the twelve base faces of HEALPix and preserves the pixel coordinates within each face, so it maps every patch onto another patch with its pixels in the same order, i.e., $i\cdot p_p+u \mapsto \pi_g(i)\cdot p_p+u$, and the embedding of the rotated map is the permuted embedding of the original one. Here $(g\cdot x)_b = x_{g^{-1}b}$, where $g^{-1}b$ is the pixel that g carries to b. Channels that hold a tangent vector in the east–north basis keep their components under rotations about z, so they transform in the same way.

B.2 CHOICE OF FRAMES AND ALIGNMENT

A token's position is a rotation carrying the north pole $\hat{n}$ to its patch center. The rotations that do this form a circle, since any one of them may be followed by a rotation about the center itself, so naming the center does not name a rotation. A frame at the center has to be chosen too, and choosing one at every point at once is a *section* of the bundle $\mathrm{SO}(3) \to S^2$.

The section matters because the encoding sees it. Of the circle of rotations carrying $\hat{n}_i$ to $\hat{n}_j$, the frames select $R_jR_i^{-1}$, and $R_i^{-1}R_j$ writes it in the frame of patch i. Re-choosing the frames by $h, h' \in \mathrm{SO}(2)_z$ carries $R_i^{-1}R_j$ to $h^{-1}R_i^{-1}R_jh'$. A difference of spherical coordinates, by contrast, is not a rotation at all, and does not even determine the distance between two points. Section 4.3 builds the positional encoding out of $R_i^{-1}R_j$.

Equation (3) makes one such choice: it turns every frame until its axes point south and east, along the local meridian and parallel, so that every patch is read with the same north. We call this the *north-aligned* section. It is a canonicalization and not a consequence of the geometry. Nothing about the sphere prefers it, no section is continuous on the whole sphere by the hairy ball theorem, and this one is undefined at the poles, where the longitude is; no patch center lies at a pole. What recommends it is that it is the familiar frame of spherical coordinates, and that a rotating planet has a preferred axis for it to agree with.

The block-aligned section. A common north is a choice, so it can be measured. The *block-aligned* section declines it, replacing $R_i = \mathcal{R}(\theta_i, \phi_i)$ by,

$$R_i^{\text{blk}} \;=\; R_i\,R_z(\tau_i), \tag{14}$$

where τ_i is the angle between the axes of patch i's own HEALPix block and the meridian at its center, and R_z is a rotation about the polar axis. Each patch is then read in the frame its own block gives it, rather than in a frame shared with its neighbors. The construction around it is untouched: the four orientations of the frame lift are still $R_i^{\text{blk}}R_z(k\pi/2)$, the resampling weights of (15) are unchanged, the token count and the cost are identical, and (14) reduces to the north-aligned section at $\tau_i = 0$.

This functions as eliminating the prior of using the north pole, and thus makes the method approximately $SO(3)$ equivariant even where the grid doesn't allow exact symmetry, in that the polar axis is no longer privileged. Under this change, the relative rotation becomes $R_z(-\tau_i)R_i^{-1}R_jR_z(\tau_j)$, which is (14) applied at each end, so the direction from one patch to another and the distance between them are what they were, and the latitude of a patch reaches the model as an input field either way (Appendix H.2). What changes is that the frames stop agreeing on a common north, and thus that positional information isn't embedded in the frames themselves. Under (3), two patches in the same geometric relationship give the same relative rotation wherever on the sphere they sit. Under (14) they do not, because their two blocks are tilted differently, and the model must account for the tilt instead of being handed it, and thus there is no privleged axis.

Both sections are exactly equivariant to the same group, but the north pole method breaks the symmetry deliberately where it isn't exact and block-frames does not. Because τ_i is a property of the fixed blocks, it is invariant under the quarter turns, which permute those blocks and therefore leave (14) equivariant, and F gives the flipped patch the same τ, so Proposition 3 and its D_4 carry over unchanged. What the north-aligned section adds is a common north in every token's frame. The lift itself asks for no such thing: a frame is already attached to every patch, and (14) reads the patch in it.. The choice is thus a symmetry breaking prior, and the two sections are far apart everywhere rather than close with exceptions: at $N_{\text{side}} = 64$ with patches of 4×4 pixels, τ_i runs from $35.4°$ to $87.4°$ modulo $90°$ over the 3,072 patches, with a median of $45.8°$, and never comes within $2.6°$ of a multiple of a quarter turn.

What the fiber absorbs, and what it does not. A section is a gauge, and the frame lift is built to make the gauge immaterial. It succeeds, but only up to the fiber.

Proposition 4 (The frame lift is locally gauge invariant)**.** *Replace the section at a single patch by $R_i \mapsto R_i r$ for any r in the fiber C_4, leaving every other patch alone. The four tokens over that patch and their elements are permuted among themselves, every relative rotation in* (9) *and* (11) *is unchanged, and so is the output. The frame lift is therefore exactly invariant under an independent C_4 at every patch.*

Proof of Proposition 4. The tokens over patch i are $\{R_i R_z(k\pi/2)\}_{k<4}$, and right multiplication by $r = R_z(m\pi/2)$ maps that set to itself by $k \mapsto k + m$. The content of a token is the patch read in the frame of its own element, so it is carried along with the element. Every relative rotation between a pair of tokens is therefore unchanged up to that relabeling, and attention is a sum over tokens. □

in the limit, the fiber in the continuous limit, where it is arbitrarily fine, the model is approximately $SO(3)$ equivariant. However, for frame lift, we only are exact where the HEALpix grid admits exact rotations onto itself, or D_4, but block-frames and north pole isolate the benefit of a symmetry breaking prior on physical signals, like weather H.7.

B.3 Non-spatial tokens

A CLS token carries no position. We give it the projector P of (7) in place of a rotation, i.e., its query and key are multiplied by P. By (7), $P\rho(R_j) = P$, so CLS matches the patch tokens through their n_0 trivial channels alone, and a larger n_0 lets it match them more selectively. With value rotation, the CLS output of a head is $\sum_j a_{\text{CLS},j} Pv_j$, where $a_{ij} = \text{softmax}_j(s_{ij})$ denotes the attention weights, so CLS aggregates only the n_0 trivial channels of every value, whereas without value rotation it aggregates all d_h of them.

Proposition 5 (CLS tokens are G-invariant)**.** *With the projector P as its position matrix, the logits involving* CLS *are $s_{\text{CLS},\,j} = q_{\text{CLS}}^\top P k_j/\sqrt{d_h}$ and $s_{j,\,\text{CLS}} = q_j^\top P k_{\text{CLS}}/\sqrt{d_h}$, neither of which depends on the rotation of any token. Proposition 2 holds with a* CLS *token, which Π leaves in place, and the* CLS *embedding is G-invariant.*

Proof of Proposition 5. Both expressions follow from (7). Under $g \in G$, the key and the value of patch j move to slot $\pi_g(j)$ unchanged, so the logits permute with the tokens and the CLS output is a permutation-invariant sum. With value rotation, we also apply P to the CLS value and to the CLS output, so that every factor of (11) between CLS and a patch token, $P\rho(R_j)$ or $\rho(R_i)^\top P$, equals P by (7) and depends on no rotation either. □

C Frame lift in detail

This appendix gives the parts of the construction of the frame lift (Sections 4.2 and 4.5) that the main text summarizes.

C.1 Rotated patches

To form the tokens of patch i, we need the values of the field at the pixel positions of the patch turned by k quarter turns about its center, and the HEALPix grid provides these values exactly only for some k. A HEALPix patch is a rhombus rather than a square. The p_p pixels of a patch form a

$\sqrt{p_p} \times \sqrt{p_p}$ array in the coordinates of its base face, and we write γ for the quarter turn of this array. The half turn γ^2 of the array moves most pixel centers close to where a geometric half turn about the patch center would place them, with a small error for patches in the equatorial zone $|\cos\theta| < 2/3$ and a larger one in the polar caps, and γ^2 is exactly the relabeling that F induces within every patch. The quarter turn γ matches a geometric quarter turn much less closely, because the patch is not square. We therefore define orientation 2 by the pixel permutation γ^2, and orientation 1 by conservative resampling, in which every entry of the patch receives the mean of the field over the quarter-turned pixel. More precisely, we compute entry u of orientation 1 of patch i as,

$$r_{iu} = \sum_b w_{iub}\, x_b, \qquad w_{iub} = \frac{\left|R_i R_z(\pi/2) R_i^{-1}\, \Omega_{iu} \cap \Omega_b\right|}{|\Omega_{iu}|}, \tag{15}$$

where r_{iu} is the resampled value, x_b is the value of the field at pixel b, Ω_b is the region of pixel b, Ω_{iu} is the region of pixel u of patch i, $R_i R_z(\pi/2) R_i^{-1}$ is the quarter turn about the center of patch i, $|\cdot|$ denotes area, and w_{iub} is the fraction of the quarter-turned region that overlaps pixel b. We define orientation 3 as orientation 1 relabeled by γ^2, which keeps the half turn F exact. The weights of every entry sum to one and are not learned. The resampling commutes exactly with the rotations that permute the pixels, i.e., with the quarter turns about z and the half turn F. We compute the weights for one pixel per orbit of D_4 and transport them by the pixel permutations, so the weights that we estimate numerically (Appendix H.2) commute exactly as well. The limitation of this choice is that most entries average several pixels, so orientations 1 and 3 carry less fine-scale detail than orientations 0 and 2. Appendix H.2 quantifies both the geometric error of γ^2 and this smoothing for the grid we use.

The frame lift expresses every channel that is a tangent vector in the local east–north basis of each pixel, turned by k quarter turns for orientation k (Section 4.2). This basis is the frame of $R_i R_z(\pi/2)$ rather than of R_i, and the fixed quarter turn between the two is a linear map that the patch embedding absorbs.

C.2 Tokens and cost

WRAP itself is cheap. It multiplies the query and the key of a token by the block-diagonal orthogonal matrix $\rho(R)$ of (5), which costs $n_0^2 + \sum_{\ell=1}^{L}(2\ell+1)^2$ multiply-accumulate operations per token and head, against Dd_h for the projection that formed that query, where D is the width of a token and d_h the head dimension. For the weather models, with $n_0 = 1$, $L = 7$, $d_h = 64$ and $D = 1024$, this is 680 against 65,536, i.e., 1%. Value rotation adds two further multiplications of the same size. The rotations are the same in every layer and are stored once (Appendix D).

The cost of the frame lift is not cheap, and comes from its token count. The encoder processes $4N$ tokens with rotations R_{ik}, where N is the number of patches. We apply every linear layer to each token separately, with weights shared across orientations, so linear layers never mix the four frames and only attention does. We compare the cost with two alternatives. First, relative to the spherical-coordinate lift, which processes N tokens of width D, the linear layers cost four times as much, $4ND^2$ against ND^2 operations with the same D^2 parameters, and attention sixteen times as much, $16N^2D$ against N^2D. Second, a model that stores the four orientations of a patch as the $4D$ channels of one token, i.e., in the regular representation of C_4, as Platonic Transformers do for their groups (Islam et al., 2026), processes N tokens of width $4D$. Its C_4-equivariant linear layers have $4D^2$ parameters. Applied densely, they cost $16ND^2$ operations, four times ours. In the Fourier basis of C_4, in which the regular representation splits into two one-dimensional irreducible representations and one two-dimensional one of complex type, they cost $D^2 + D^2 + 4D^2$ per token, i.e., $6ND^2$, one and a half times ours. Its attention costs $4N^2D$, a quarter of ours. Storing the four orientations as tokens therefore saves in the linear layers and pays fourfold in attention. Appendix H.2 (*Cost*) measures the cost per training step.

C.3 Output and choice of orientations

Without mapping the prediction of orientation 2 back to the frame of orientation 0, the mean would combine pixels on opposite sides of the patch center and cancel vector channels, whose components change sign under a half turn. We average predictions rather than features because a change of frame acts on predictions by a known map, i.e., a pixel permutation and a rotation of vector channels,

whereas the features of tokens with different orientations are related by no fixed map. If the grid provided the four orientations of every patch exactly, the natural output would be the mean over all four, i.e., the projection from functions on the four frames back to the sphere that a group convolution uses. On HEALPix, only orientations 0 and 2 are exact pixel permutations, which is why the output uses them alone.

Choice of orientations. Two requirements determine which orientations enter the mean. First, the output is equivariant to F only if the set of orientations it averages is closed under the relabeling $k \mapsto k + 2$ that F induces, so orientation 0 alone does not suffice. Second, the output must consist of predictions at the pixel centers, where it is defined, without resampling any prediction a second time. Orientations 1 and 3 are predicted at quarter-turned positions, and mapping them to the pixel centers would require a second resampling, which would smooth the output, would not recover the values it averaged, and would combine predictions of different effective resolution (Appendix H.2 measures this smoothing). That is, $\{0, 2\}$ is the only non-empty set of orientations that meets both requirements. The second requirement concerns accuracy rather than symmetry, since the sets $\{1, 3\}$ and $\{0, 1, 2, 3\}$ also yield exactly equivariant outputs when the predictions of all their orientations are mapped back consistently, but at positions that are not pixel centers, which we confirmed numerically. All four orientations are still trained and take part in attention, and only the choice of output is constrained.

C.4 Loss on every orientation

A loss on the output alone cannot tell whether the predictions it averages are each correct, since two predictions that err in opposite directions average to a correct output. We therefore train each of the four predictions against the target sampled at its own pixel positions. For a quadratic loss $\mathcal{Q}$, such as a weighted squared error, the loss on orientations 0 and 2 splits as,

$$\tfrac{1}{2} \sum_{k \in \{0,2\}} \mathcal{Q}\big(\hat{y}_k - y\big) \;=\; \mathcal{Q}\big(\bar{y} - y\big) \;+\; \tfrac{1}{2} \sum_{k \in \{0,2\}} \mathcal{Q}\big(\hat{y}_k - \bar{y}\big), \tag{16}$$

where $\hat{y}_k$ is the prediction of orientation k mapped back to the frame of orientation 0, $\bar{y}$ is the output, i.e., the mean of $\hat{y}_0$ and $\hat{y}_2$, and y is the target. That is, the loss on orientations 0 and 2 contains the loss of the output as its first term and a penalty on the disagreement between the predictions as its second, so it trains the output and penalizes predictions that cancel in the mean. For any convex loss, the left-hand side remains an upper bound on the loss of the output. The same failure is known for ensembles whose members are trained only through the loss of their mean (Jeffares et al., 2023). Scoring every orientation keeps each of them a prediction of the same field in its own frame, which is what the frame lift assumes when it treats the four tokens as one patch seen in four frames. Orientations 1 and 3 do not enter the output, but we still score them against the target resampled by (15), using only the terms of the loss that do not require values at the pixel centers. When the model is trained autoregressively, we apply this loss at every step, while the next step takes the output as its input. Appendix H.2 (*Loss on every orientation*) measures how far the orientations agree in our weather models.

C.5 The symmetry group

Since $F\, R_z(\alpha)\, F^{-1} = R_z(-\alpha)$, the quarter turns G and the half turn F generate the dihedral group $D_4 = \langle R_z(\pi/2), F \rangle$ of order eight (Proposition 3).

A change of frame at patch i maps the four rotations $\{R_i\, R_z(k\pi/2)\}$ onto themselves and relabels the four tokens of the patch together with their pixel positions, so every relative rotation is unchanged. The tokens are therefore equivariant to the change of frame, i.e., relabeled, and the output, which is defined by pixel positions rather than by orientation index, does not change. Unlike D_4, the gauge group C_4^N does not act on the input field or on the output, only on the labels of the tokens. It expresses that the model does not depend on which of the four frames of a patch is labeled orientation 0, as in gauge equivariant networks (Cohen et al., 2019), and D_4 acts on it by permuting the patches and, with them, the changes of frame.

The group D_4 of Proposition 3 is the largest subgroup of $\mathrm{SO}(3)$ that permutes the HEALPix pixels, since a rotation that permutes the pixels must permute the eight pixel corners at which only three

pixels meet, which are the vertices of a square prism that is not a cube and whose rotation group is D_4 (Appendix E). In the continuum, the global symmetry group becomes $\mathrm{SO}(2)_z \rtimes \langle F \rangle \cong \mathrm{O}(2)$, i.e., all rotations about the polar axis together with the half turn, all of which are proper rotations.

The half turn exchanges the hemispheres, so it is a symmetry of a learned map only when every input that distinguishes the hemispheres is supplied as a field and transformed with the other inputs.

D Real Wigner-D matrices

For a rotation $R \in \mathrm{SO}(3)$ with axis-angle vector $\eta \in \mathbb{R}^3$, let $J_x^\ell, J_y^\ell, J_z^\ell$ be the generators of the degree-ℓ representation of $\mathrm{SO}(3)$ in the complex basis $|\ell, m\rangle$, with order $-\ell \le m \le \ell$, and let U be the unitary matrix that changes basis from the real to the complex spherical harmonics of degree ℓ, i.e., $Y_{\mathbb{R}}^\ell = U^\dagger Y_{\mathbb{C}}^\ell$. Then

$$D_{\mathbb{C}}^\ell(R) = \exp\big(-\mathrm{i}(\eta_x J_x^\ell + \eta_y J_y^\ell + \eta_z J_z^\ell)\big), \qquad D^\ell(R) = U^\dagger D_{\mathbb{C}}^\ell(R)\, U \in \mathbb{R}^{(2\ell+1)\times(2\ell+1)},$$

are the complex and real Wigner-D matrices of degree ℓ, where η_x, η_y and η_z are the components of η, i is the imaginary unit and $\dagger$ is the conjugate transpose. The rotary map (5) stacks $D^1, \dots, D^L$ after n_0 trivial channels, and we compute $\rho(R_i)$ for every patch once, before training. The matrices $\rho(R_i)$ depend only on the grid, and we store them in a non-trainable buffer that is shared across heads.

E Proofs

This appendix proves the propositions and the theorem of Section 4, states the two propositions about value rotation that Section 4.4 refers to, and collects the remarks that follow from them.

Proof of Proposition 1. A rotation $R_z(\alpha)$ sends the point (θ, ϕ) to $(\theta, \phi + \alpha)$, and

$$R_z(\alpha)\, \mathcal{R}(\theta, \phi) = R_z(\alpha) R_z(\phi) R_y(\theta) = R_z(\phi + \alpha) R_y(\theta) = \mathcal{R}(\theta, \phi + \alpha).$$

For a general g, with $\hat{u}$ and $g\hat{u}$ both other than the poles, so that $\mathcal{R}$ is defined at each, both $g\,\mathcal{R}(\hat{u})$ and $\mathcal{R}(g\hat{u})$ carry $\hat{n}$ to $g\hat{u}$, so they differ by right multiplication by an element of the stabilizer $\mathrm{SO}(2)_z$. □

Proof of Proposition 2. Queries, keys and values are formed from the features of each token by shared linear maps, so ΠZ gives the permuted queries, keys and values, e.g., $q'_{\pi(i)} = q_i$. Since $R'^{-1}_{\pi(i)} R'_{\pi(j)} = R_i^{-1} S^{-1} S R_j = R_i^{-1} R_j$, the logits (9) satisfy $s'_{\pi(i)\pi(j)} = s_{ij}$, and so do the attention weights $a_{ij} = \mathrm{softmax}_j(s_{ij})$. The output (10) at $\pi(i)$ is therefore $\sum_{j'} a'_{\pi(i)j'} v'_{j'} = \sum_j a'_{\pi(i)\pi(j)} v'_{\pi(j)} = \sum_j a_{ij} v_j = \mathrm{out}_i$, summing over $j' = \pi(j)$. With value rotation, the factor $\rho(R_i^{-1} R_j)$ of (11) is unchanged for the same reason, so the same holds with v_j replaced by $\rho(R_i^{-1} R_j)\, v_j$. The concatenation of the heads and the output projection act on each token separately. □

Proof of Theorem 1. The encoder is a composition of the patch embedding, a stack of blocks and a final normalization. The patch embedding satisfies (2) (Appendix B.1, and for orientations 1 to 3 of the frame lift, Appendix C.1), every block commutes with Π_g by Proposition 2 and the remark after it in Section 4.4, also with a CLS token by Proposition 5, and the final normalization acts on each token separately. A composition of G-equivariant maps is G-equivariant. □

Frame-relative features. The features of every token are frame-relative: under $g \in G$ a token moves to $\pi_g(i)$ with its features unchanged, rather than transforming as a steerable Wigner-D vector, which the normalization and the MLP would not preserve. The rotations act only inside attention.

Proposition 6 (Averaging the four frames of a patch)**.** *Let $r_k = R_z(k\pi/2)$, and suppose that the values of the four tokens of a patch are rotated copies of one another, $v(R\, r_k) = \rho(r_k)^\top v(R)$. For a query with rotation R_i and the four tokens of patch j, with rotations $R_j r_k$,*

$$\tfrac{1}{4} \textstyle\sum_k v(R_j r_k) = P_G\, v(R_j) \qquad \textit{without value rotation},$$
$$\tfrac{1}{4} \textstyle\sum_k \rho\big(R_i^{-1} R_j r_k\big)\, v(R_j r_k) = \rho(R_i^{-1} R_j)\, v(R_j) \qquad \textit{with value rotation},$$

where $P_G = \frac{1}{4}\sum_k \rho(r_k)^\top$ is the orthogonal projection onto the vectors that quarter turns about z leave unchanged, i.e., onto the n_0 trivial channels together with the components whose order m is divisible by four in every degree ℓ.

Proof of Proposition 6. Without value rotation, $\frac{1}{4}\sum_k \rho(r_k)^\top v(R_j) = P_G v(R_j)$ by definition of P_G. With value rotation, $\rho(R_i^{-1}R_j r_k)\,\rho(r_k)^\top v(R_j) = \rho(R_i^{-1}R_j)\,v(R_j)$ for every k by (6). □

Proposition 7 (How much value transport can change, and when)**.** *Fix a head and a query token i, and suppose the degree-1 block $v_j^{(1)}$ of the value of token j holds a tangent vector at that token's center expressed in its own frame, as in Proposition 6. Let $\theta_{ij} \in [0,\pi]$ be the rotation angle of $R_i^{-1}R_j$. Then the difference between the head output with and without value rotation, restricted to degree 1, satisfies,*

$$\Big\| \textstyle\sum_j a_{ij}\big(D^1(R_i^{-1}R_j) - I\big)\, v_j^{(1)} \Big\| \;\le\; \sum_j a_{ij}\; 2\sin\big(\theta_{ij}/2\big)\, \big\|v_j^{(1)}\big\|, \tag{17}$$

where $a_{ij} = \mathrm{softmax}_j(s_{ij})$ are the attention weights, with equality for a single attended token whose vector is orthogonal to the axis of $R_i^{-1}R_j$. The difference therefore vanishes as $\theta_{ij} \to 0$, and for two tokens of one patch of the frame lift, whose relative rotation is $R_z(k\pi/2)$, the factor $2\sin(\theta_{ij}/2)$ is $\sqrt{2}$, 2 and $\sqrt{2}$ for $k = 1, 2, 3$.

Proof of Proposition 7. Let $R \in \mathrm{SO}(3)$ have angle θ and axis $\hat{a}$, and split $x \in \mathbb{R}^3$ as $x = x_\parallel + x_\perp$ along and across $\hat{a}$. Then $Rx_\parallel = x_\parallel$ and R turns $x_\perp$ by θ within its plane, so $\|(R-I)x\| = \|(R-I)x_\perp\| = 2\sin(\theta/2)\|x_\perp\| \le 2\sin(\theta/2)\|x\|$, with equality iff $x_\parallel = 0$. As in the proof of Proposition 8, $D^1(R) = \Sigma R \Sigma^\top$ with Σ a permutation matrix, so $\|(D^1(R)-I)y\| = \|(R-I)\Sigma^\top y\|$ and the same bound holds in the degree-1 block. Applying it termwise to $\sum_j a_{ij}(D^1(R_i^{-1}R_j) - I)v_j^{(1)}$ and using $a_{ij} \ge 0$ with the triangle inequality gives (17). The stated values follow from $\theta = k\pi/2$. □

Why the lift decides whether value rotation is needed. The two propositions above separate the two regimes, and the separation is a property of the lift rather than of the data. In the frame lift, the four tokens of one patch sit at relative rotations $R_z(k\pi/2)$ *whatever* their position on the sphere, so the mismatch in (17) is fixed at a quarter or a half turn and cannot be made small by attending locally, and Proposition 6 shows the loss is then total rather than merely bounded: a head that weights the four siblings equally returns P_G of the value, which annihilates both tangential components of a vector. A lift with one frame per patch has no sibling tokens, and the only mismatch a head sees is θ_{ij} between distinct patches, which tends to 0 as two patches approach each other, so a head that places its weight on nearby patches sees frames that nearly agree and transport is close to the identity. Value rotation is therefore structural for a multi-frame lift and optional for a single-frame one. That is what Table 7 measures: and the difference is about a percent on the 850 hPa wind for the frame lift, and for the spherical-coordinate lift it makes no meaningful difference.

Proposition 8 (What value rotation changes)**.** *(i) Value rotation does not change the symmetry of the model, i.e., Proposition 2 holds with and without it, and so do Theorem 1 and Proposition 3. (ii) Value rotation acts as the identity on the trivial channels of every value, so content in them, such as an invariant scalar, is combined in the same way with and without it. (iii) If the degree-1 component of the value of token j holds a tangent vector in the frame of token j, value rotation expresses it in the frame of the query, so a head adds vectors written in one frame. Without it, the head adds components written in different frames, and when the four tokens of a patch hold the same vector, a head that averages over their four frames removes both of its tangential components (Proposition 6).*

The model does not force a tangent vector into the degree-1 component, so part (iii) states what value rotation lets a head represent, not what a trained head does.

Proof of Proposition 8. (i) The proof of Proposition 2 covers both variants, and the proofs of Theorem 1 and Proposition 3 use the blocks only through it. For a quarter turn $g \in G$, $R' = R$ and $S = g$ by Proposition 1. For the frame lift, the half turn satisfies $F\,R_i R_z(k\pi/2) = R_{i'} R_z((k+2)\pi/2)$ for every patch i, where i' is the patch whose center is $F\hat{n}_i$, so $R' = R$ and $S = F$, and a change of frame at one patch relabels its four tokens together with their rotations, so $R'_{\pi(i)} = R_i$ and $S = I$ (proof of Proposition 3 below).

(ii) By (7), $P\,\rho(R_i^{-1}R_j)\,v_j = P\,v_j$, so for the same attention weights, the trivial channels of the output are $P\,\mathrm{out}_i = \sum_j a_{ij}\,Pv_j$ with and without value rotation.

(iii) The real spherical harmonics of degree 1 and orders $m = -1, 0, 1$ are proportional to y, z and x, so $D^1(R) = \Sigma R \Sigma^\top$ for every $R \in \mathrm{SO}(3)$, where Σ is the permutation matrix that maps $(x, y, z)^\top$ to $(y, z, x)^\top$. Let $\xi \in \mathbb{R}^3$ be a tangent vector at the center $\hat{n}_j$ of the patch of token j, and let $\xi_j = R_j^{-1}\xi$ be its components in the frame of token j. Since $R_j\hat{n} = \hat{n}_j$, $\xi_j \perp \hat{n}$. Suppose that the degree-1 component of v_j is $c_j\,\Sigma\xi_j$ for a scalar c_j. With value rotation, the degree-1 component of $\rho(R_i^{-1}R_j)\,v_j$ is $D^1(R_i^{-1}R_j)\,c_j\Sigma\xi_j = c_j\,\Sigma R_i^{-1}\xi$, i.e., the same vector written in the frame of token i, so the head adds the vectors $c_j\xi$ in $\mathbb{R}^3$ and writes their sum in the frame of the query. This is a rotation, not a projection onto the tangent plane at the query, so for distant patches the sum can have a component along $R_i^{-1}\hat{n}_i = \hat{n}$. Without value rotation, if every token j holds a vector $\xi^{(j)}$ in this way, the degree-1 component of (10) is $\Sigma \sum_j a_{ij}\,c_j\,R_j^{-1}\xi^{(j)}$, a sum of components written in the different frames R_j. Now let the four tokens of patch j, with frames $R_j r_k$ and $r_k = R_z(k\pi/2)$, hold the same vector ξ with the same scalar c_j. Their components are $(R_j r_k)^{-1}\xi = r_k^\top \xi_j$, so the degree-1 component of $v(R_j r_k)$ is $\Sigma r_k^\top \Sigma^\top\,(c_j\Sigma\xi_j) = D^1(r_k)^\top$ times that of $v(R_j)$, which is the hypothesis of Proposition 6 in degree 1. A head that weights the four tokens equally therefore returns P_G applied to this component without value rotation. In degree 1, $P_G = \Sigma\left(\frac{1}{4}\sum_k r_k^\top\right)\Sigma^\top = (\Sigma\hat{n})(\Sigma\hat{n})^\top$ keeps only the order $m = 0$, i.e., the component along $\Sigma\hat{n}$, which vanishes since $\xi_j \perp \hat{n}$, so both tangential components are removed. With value rotation, the head returns $c_j\,\Sigma R_i^{-1}\xi$ by Proposition 6, as desired. □

What value rotation changes. On a line all positions share one frame, so a token needs no change of frame before another token adds it, whereas on the sphere the frames of neighboring patches are rotated relative to each other. Value rotation does not change the group to which a model is equivariant (Proposition 8, part (i)). What it changes is how a head combines content across frames. In the frame lift, the four tokens of a patch hold the same content in frames that differ by quarter turns. When their values are rotated copies of one another, a head that spreads its weight evenly over those four tokens averages them to the value of the patch in the frame of the query with value rotation, and to their projection P_G onto the components that quarter turns about z leave unchanged without it (Proposition 6). Those are the components whose order satisfies $m \equiv 0 \pmod 4$, which for a head of dimension 64, i.e., degrees $\ell \leq 7$, is 16 of its 64 dimensions. Under the hypothesis of Proposition 6, content that depends on orientation, such as a tangent vector or the gradient of a scalar field, transforms under quarter turns as that content does, so it lies in the orders $m \not\equiv 0 \pmod 4$ and vanishes under P_G. Content that quarter turns leave unchanged, such as the mean of a scalar field or the mean vorticity of the wind, survives in the remaining quarter of the representation, at the cost of width in that path. For a vector channel the lost part is the defining content, the direction of the vector, whereas a scalar field loses only its orientation-dependent structure. The model imposes neither rotated copies nor equal weights, and it does not force a tangent vector into the degree-1 component, so this argument motivates value rotation rather than predicting its effect on accuracy, which Appendix H.7 measures for scalar fields and for the wind.

Proof of Proposition 3. A quarter turn $g \in G$ maps R_{ik} to $R_{\pi_g(i)k}$ by Proposition 1. The half turn satisfies $F\,\mathcal{R}(\hat{u}) = \mathcal{R}(F\hat{u})\,R_z(\pi)$ for every $\hat{u} \in S^2$ other than the poles, so it maps token (i, k) to token $(i', k + 2 \bmod 4)$, where i' is the patch whose center is $F\hat{n}_i$. The HEALPix pixel set is invariant under both rotations, and the resampling (15), with orientation 3 defined as orientation 1 relabeled by γ^2, commutes with both. For the frame lift, both rotations therefore permute the tokens and preserve every relative rotation. They also leave the content of every token unchanged. For F, this holds because F negates the vector channels and the additional half turn $R_z(\pi)$ in the frame of token $(i', k + 2)$ undoes this negation. Hence the argument of Proposition 2 applies, with and without value rotation, since the attention weights and the factor of (11) depend on the tokens only through relative rotations. The final layer acts on each token separately, so the prediction of token $(i', k + 2)$ on $F \cdot x$ equals that of token (i, k) on x. Since $\{0, 2\}$ is closed under $k \mapsto k + 2$ and orientation 2 is mapped back by γ^2 and a half turn of its vector channels, $\bar{y}(F \cdot x) = F \cdot \bar{y}(x)$. A change of frame at a single patch j, $R_j \mapsto R_j R_z(\pi/2)$, permutes the four tokens of j together with their rotations and pixel positions, so every relative rotation and the prediction of every token are unchanged. The relabeling carries the two readings that lie at the pixel centers with it, from orientations $\{0, 2\}$ to $\{3, 1\}$, and the output is the mean over that pair, so it is relabeled with the tokens and does not change, as desired. Finally, a rotation that permutes the pixels maps pixel

Table 3: **Masked spherical pretraining on the Manifold40 radar dataset.** Reconstruction error on held-out masked patches. Lower is better. standard error. Both variants of our method outperfrom the two competitive baselines.

Model	Position encoding	Recon. MSE ↓
Baselines		
ViT (Dosovitskiy et al., 2021)	Learned	0.2310
Attention on the Sphere (Bonev et al., 2025b)	Spectral	0.2788
Ours		
SPHERICAL-COORDINATE LIFT	Spherical RoPE	0.1457
FRAME LIFT	Spherical RoPE	0.1133

corners to pixel corners and preserves the number of pixels that meet at each. Exactly eight corners, at $z = \pm 2/3$ and $\phi \in \{0, \pi/2, \pi, 3\pi/2\}$, are shared by three pixels rather than four. They are the vertices of a square prism that is not a cube, whose rotation group is D_4, so no subgroup of $\mathrm{SO}(3)$ larger than D_4 permutes the pixels. □

F RADAR BENCHMARK: FURTHER DETAILS

F.1 TRAINING DETAILS

We first resample these radar profiles from lat-lon grid onto a HEALPix grid at $\texttt{nside} = 64$ and normalize each sample by the training-set mean and standard deviation. We split the training set $90/10$ into train and validation, and evaluate on a held-out test set.

As done in He et al. (2022), we train the classification model in two stages. First, we pre-train the encoder by reconstructing the input radar signal from a 75% masked input signal using a mean squared error loss. After pre-training we discard the decoder and attach a classification head to the pre-trained encoder. We pretrain with AdamW (Loshchilov & Hutter, 2019) ($\beta = (0.9, 0.95)$, weight decay 0.05) at a learning rate of 6×10^{-3} and an effective batch size of 1024, for at most 800 epochs, with a linear warmup over the first 40 epochs followed by a cosine decay to zero. We clip gradients at global norm 4.0. For the linear probe we freeze the encoder and train only the head, which is a layer normalization, dropout of 0.1, and a linear map to the 40 classes. We again use AdamW (weight decay 0.05), at a learning rate of 4×10^{-2} and an effective batch size of 1024, for 50 epochs with 5 warmup epochs, clipping gradients at 1.0. The larger learning rate is affordable because only the head is trained. We used Nvidia V100 GPUs for training.

F.2 RADAR SIGNAL RECONSTRUCTION RESULTS

This section provides supplementary results that highlight the performance of our masked spherical SSL models on the Manifold40 radar dataset. Table F.2 presents the reconstruction error for both variants of our approach: the spherical-coordinate lift and the frame-lift. We also include the performance of two competitive baselines: the standard Vision Transformer (ViT) (Dosovitskiy et al., 2021) and Attention on the Sphere (Bonev et al., 2025b) and demonstrate that our method outperforms both approaches, with the frame-lift variant achieving the lowest reconstruction error. Representative examples of reconstructed radar signatures from the ViT baseline and our model alongside the corresponding ground truth are shown in Figure 6.

G SPHERICAL SEGMENTATION TRAINING DETAILS

We split the 1413 panoramas $95/2.5/2.5$ into train, test and validation with a fixed seed reproducing the split of Bonev et al. (2025b) exactly so that their published numbers are a fair reference for ours. We also exclude from the loss and from the metrics. Each channel is normalised by the quadrature-weighted mean and standard deviation of the training split

We train for 200 epochs with AdamW(Loshchilov & Hutter, 2019) (weight decay 0.1) at a learning rate of 5×10^{-4}, cosine-decayed to 1×10^{-6} with no warmup, at a batch size of 8, clipping gradients

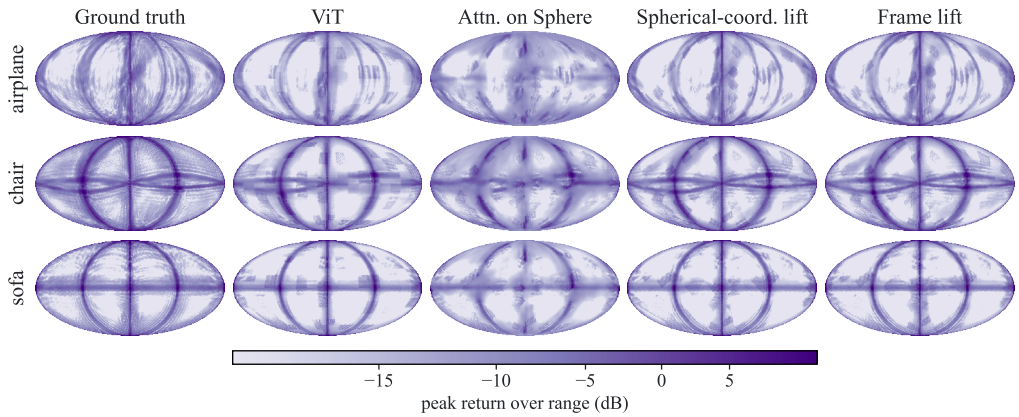


Figure 6: MAE reconstructions from the four pretrained encoders on three Manifold40 test samples. Each row is one object. The first column is the ground-truth signal, and the remaining columns are reconstructions by the ViT, Attn. on Sphere, the spherical-coordinate lift and the frame lift. Every model sees the same mask, which hides 75% of the 192 HEALPix patches ($N_{\text{side}} = 4$) and is identical across samples. Hidden regions show the model's prediction and visible regions the input. Maps show the peak return over range in dB in Mollweide projection, on one colour scale shared by all panels. The frame lift has the lowest reconstruction error on the hidden region for all three objects.

at global norm 4.0. The loss is a quadrature-weighted cross entropy on the HEALPix grid the model predicts on; since HEALPix pixels are equal area, the quadrature weights reduce to a uniform $1/\texttt{npix}$. We report mean intersection-over-union and accuracy in both spaces. We used a single Nvidia B200 GPU for training.

H Weather benchmark: further details

H.1 Setup

What the benchmark tests. Weather forecasting tests a partial symmetry that is known exactly. The equations of motion are invariant under rotations about the polar axis, the reflection through the equatorial plane and, provided that the Coriolis parameter changes sign under it, the half turn F of Section 5.3. That parameter enters our models only through the input $\sin(\text{latitude})$, which F transforms together with the other inputs, so F is a symmetry of the forecast map. Local isotropy about the vertical holds only approximately, since the latitudinal gradient of the Coriolis parameter, orography and the land–sea contrast all break it. The benchmark therefore tests whether building in the symmetries of the grid, including the per-patch changes of frame of the frame lift, helps when the symmetry about the polar axis is exact and the symmetry-breaking fields are supplied as inputs. Averaging over step sequences lowers RMSE as ensemble averaging does, so we report a deterministic forecast and the mean over K step sequences, each against baselines of the same kind (Appendix H.4).

Data. We use ERA5 (Hersbach et al., 2020) at 6-hourly resolution, as regridded to $1.5°$ by WeatherBench 2 (Rasp et al., 2024), and resample it to HEALPix (Górski et al., 2005) with $N_{\text{side}} = 64$, i.e., 49,152 pixels. The state has 83 channels: geopotential, temperature, horizontal wind, specific humidity and vertical velocity at 13 pressure levels, as well as 2 m temperature, 10 m wind, mean sea-level pressure and 6-hour precipitation. The static inputs are orography, the land–sea mask and $\sin(\text{latitude})$, which is how the Coriolis parameter reaches the model (*What the benchmark tests*). We train on 1979–2018, select checkpoints on 2019, and test on 2020.

Evaluation. We map forecasts back to the $1.5°$ grid (240×121) by first-order conservative regridding, estimated with 64×64 sample points per cell, and score them against ERA5 under the WeatherBench 2 protocol, i.e., from all 732 initializations at 00 and 12 UTC in 2020 with the published metric definitions. We report latitude-weighted RMSE and anomaly correlation coefficient

(ACC) for z500, T850, T2m, specific humidity at 700 hPa (q700), 10 m zonal wind (U10) and mean sea-level pressure (MSLP). Since RMSE alone rewards smooth forecasts, we also report zonal energy spectra. We compute RMSE and ACC with our own implementation, which reproduces the published RMSE and ACC of IFS HRES for z500, T850, q700, T2m and MSLP exactly at 1, 3, 5 and 10 days (Appendix H.10). Every result for 2020 on skill and physical consistency, in the main text and in Appendices H.5–H.12, follows this protocol. The 2019 scores and the diagnostics elsewhere in the appendix use the subsets of initializations that each table or paragraph states.

Using 2019 as the validation year. The stage-2 checkpoint, the step length of the deterministic forecast and the perturbation amplitude of each ensemble are chosen on 49 initializations of 2019, every 15th, with checkpoints scored as 6-hour rollouts. Nothing measured on 2020 enters any of those three choices. The WeatherBench 2 protocol verifies late-year forecasts against the year that follows, so the last of these initializations is verified, at leads from 5 days on, against ERA5 of the first six days of 2020. The training recipe, which every model shares, was fixed in preliminary runs, some of them also scored on 2020.

Precision for training and benchmarking. Training uses bfloat16 mixed precision. Every forecast behind the results above is computed in bfloat16 autocast, the same for every row, and so are the 2019 forecasts that choose the step length. The 2019 scores that select the stage-2 checkpoint, and the diagnostics, are computed in float32 unless stated otherwise.

Weather modeling using WRAP-former. Every model builds on the training setup of Stormer (Nguyen et al., 2024), which reaches strong WeatherBench 2 scores with a plain transformer at coarse resolution and conditions one model on the forecast interval, but with significant changes to integrate our methodology for physical spherical signals. We keep its interval sampling, its prediction target and its conditioning through adaptive layer normalization, and change four components: we embed each HEALPix patch, by default $N = 3{,}072$ patches of 16 pixels, with one linear map; position enters through the encodings compared below; queries and keys are formed without the interval conditioning and every head gains a key and value fixed at zero (Appendix H.13); and the objective gains a spectral term. All models share a 24-block transformer of width $D = 1024$ with 16 heads and a two-layer MLP head. Appendix H.3 gives every change with its reason.

Compared models. The models differ only in the positional encoding, the lift, value rotation and the augmentation, except two controls at twice the width (Table 6). Every model with Wigner-RoPE uses value rotation, which Appendix H.7 removes. Both lifts are also trained with the second stage below, and those two models are WRAP-former-Frame and WRAP-former-Spherical; a model named as a lift is the first-stage model of the controlled comparison.

Objective. Atmospheric power falls steeply with degree ℓ, so plain squared error concentrates its gradient on the largest scales. We therefore train every model with a spectral objective, which adds a spectral term to the squared error on the grid. That is, all models minimize,

$$\mathcal{L} = (1-\lambda)\,\mathcal{L}_{\text{grid}} + \lambda\,\mathcal{L}_{\text{spec}}, \qquad \mathcal{L}_{\text{spec}} = \frac{1}{C}\sum_{c} w_c\,\kappa_c \sum_{\ell \le 64} \frac{\sum_m |\hat{e}_{c,\ell m}|^2}{\sigma_{c,\ell}^2 + \epsilon_c}, \tag{18}$$

where $\lambda = 0.5$ is the weight of the spectral term, $C = 83$ is the number of predicted channels, w_c is the weight of channel c, from the pressure-level and variable weights of GraphCast (Lam et al., 2023), which already normalize by the variance of the 6-hour increment, shared between the two components of each horizontal wind and normalized to mean one, and $\mathcal{L}_{\text{grid}}$ is the mean over channels of the w_c-weighted squared error over the equal-area HEALPix pixels. In the spectral term, $\hat{e}_{c,\ell m}$ is the spherical-harmonic coefficient of degree ℓ and order m of the error in channel c, $\sigma_{c,\ell}^2$ is the climatological power of the 6-hour increment of channel c at degree ℓ, used for all three intervals, ϵ_c is 5% of the peak power of channel c, and κ_c normalizes the spectral term of channel c so that an error with the climatological spectrum scores as it does under $\mathcal{L}_{\text{grid}}$. Within each channel, the spectral weight $1/(\sigma_{c,\ell}^2 + \epsilon_c)$ depends only on ℓ, so the objective is invariant under rotations about the polar axis and the half turn F. For the frame lift, $\mathcal{L}_{\text{grid}}$ applies to all four orientations and $\mathcal{L}_{\text{spec}}$ only to orientations 0 and 2, whose predictions lie at the pixel centers. Each orientation carries one quarter of the loss (Appendices C.4 and H.2), so the grid term of the frame lift carries weight

$1-\lambda$, as for the other models, while its spectral term carries $\lambda/2$, half the weight it has for the other models. Unlike AMSE (Subich et al., 2025), which changes the optimum under forecast uncertainty to recover spectral amplitude, the spectral term only reweights the error by degree. Appendix H.14 compares the objective with plain squared error.

Training. We train in two stages. In the first, we train on single forecast steps with AdamW and a constant learning rate followed by a cooldown, i.e., a final phase in which the learning rate decays to zero (Hägele et al., 2024). Because the learning rate is constant before the cooldown, we can start a cooldown from any saved checkpoint without retraining. We score controlled comparisons and ablations after a cooldown started from the checkpoint at 48k steps, or at 24k steps for the one run that was halted before 48k steps. We report one run per configuration, since a single run of the frame lift to the end of its scored cooldown costs about 78 GPU-hours. The ablations of Stormer (Nguyen et al., 2024) and ArchesWeather (Couairon et al., 2024) also report single runs without error bars. Appendix H.6 measures how far two runs of one configuration differ. The first stages of WRAP-former-Frame and WRAP-former-Spherical, the two models of Table 8, are scheduled for 288k steps, i.e., 9.2M samples against the 9.6M of GraphCast (Lam et al., 2023) (Appendix H.2, *Stage 1*). Both complete that schedule. In the second stage, a forecast feeds the model its own outputs, so we fine-tune both models on autoregressive rollouts with the curriculum of GraphCast, in which the rollout grows by one forecast step every 1,000 updates, here from 2 to 10 steps, and the loss (18) applies at every step. Appendix H.2 gives the full recipe and the cost of the frame lift.

H.2 Experimental details

Input normalization. Each channel is standardized by its own mean and standard deviation, except the two horizontal components of each wind, which share one constant. This matters only for the block-aligned frames of Table 7, whose angle to the meridian is never a multiple of a quarter turn: their transport mixes the two components, and with separate constants the mixture is not the transported vector. The two orientations a north-aligned lift decodes, 0 and 2, turn the pair by 0 or $180°$, which is a sign flip and therefore exact under either convention.

Optimization. We use AdamW with learning rate 10^{-3} (5×10^{-4} for the two runs at width 2048; Appendix H.13), $\beta=(0.9, 0.95)$, weight decay 0.1 on weight matrices and conditioning biases, and 1,000 warm-up steps, followed by a constant learning rate and a cooldown over the last 20% of the steps after warm-up, in which the learning rate decays to zero as $1-\sqrt{t}$, where t is the elapsed fraction of the cooldown (Hägele et al., 2024). Gradient-norm clipping is set to 1.0, and we train in bfloat16 mixed precision. The effective batch size is 32 on 1 to 8 GPUs, with gradient accumulation making up the difference.

Stage 1. We train on single forecast steps whose interval is drawn uniformly from $\{6, 12, 24\}$ h. A cooldown started from the checkpoint at s_0 steps runs to step $1000+(s_0-1000)/0.8$, so that it spans the last 20% of the steps after warm-up, e.g., from 48k to 59,750 steps. Controlled comparisons and ablations start cooldowns from the checkpoints at 24k and 48k steps and are scored after the second. A run that diverges before 48k steps is scored after the cooldown started at 24k steps. The frame lift and the spherical-coordinate lift train on the full schedule of 288k steps, whose last 20% is the cooldown of the run itself, so the two are matched in steps. The frame lift also starts cooldowns from its checkpoints at 24k, 48k, 96k, 144k and 192k steps, which we score on 49 initializations of 2019 to follow training. Rotation augmentation draws one rotation per sample from a fixed set. Rotations that are grid symmetries act as exact pixel permutations, and all other rotations are applied by conservative resampling. The D_4 augmentation draws from its 8 elements, i.e., the 4 quarter turns about the polar axis, each with and without the half turn, all of which are exact pixel permutations. The $\mathrm{O}(2)$ augmentation draws from 64 elements, i.e., 32 rotations about the polar axis, each with and without the half turn. The $\mathrm{SO}(3)$ augmentation draws from the identity and 63 random rotations.

Stage 2: curriculum. A model trained on single steps reads analyzed states, whereas a forecast feeds it its own outputs, whose errors compound with lead time. We therefore fine-tune the frame lift and the spherical-coordinate lift on autoregressive rollouts, starting from the end of the first stage, as is standard. We follow the curriculum of GraphCast and AIFS (Lang et al., 2024). The rollout starts at 2 forecast steps and grows by one step every 1,000 updates until it reaches 10 steps, and then

continues at 10 steps, for 12,000 updates in total at batch size 32. This exceeds the 11,000 updates of GraphCast, which reach 12 steps and whose increment AIFS matches without stating a total. The loss is (18) averaged over the forecast steps of the rollout, each against its own target, and gradients flow through the full rollout. As in Stormer (Nguyen et al., 2024), every rollout draws one interval from $\{6, 12, 24\}$ h and uses it at every step, since Stormer found that changing the interval within a rollout destabilizes training. A rollout of S steps therefore spans $6S$, $12S$ or $24S$ hours. We stop at 10 steps rather than the 12 of GraphCast because 10 steps of 24 hours reach 240 hours, the longest lead time we score, and GraphCast finds that a longer training horizon trades accuracy at short lead times for accuracy at long ones.

Stage 2: optimization and loss. The learning rate is constant at 5×10^{-6} after 200 warm-up steps, the rate of the first rollout stage of Stormer, which is 8 to 50 times the constant rates of GraphCast, FuXi and AIFS. Weight decay and clipping are as in stage 1. We keep the objective of stage 1 at every step, since every step applies the same one-step map, whose error is naturally measured against the increment over one interval regardless of the position of the step in the rollout. For the frame lift, the loss scores every orientation at every step of the rollout, as in stage 1, while the next step takes the output as its input.

Stage 2: checkpoint selection. We save a checkpoint every 1,000 updates and score the stage-1 starting point and the checkpoints at 4k, 7k and 9k updates on 49 initializations of 2019, as 6-hour rollouts to 10 days. We run the full curriculum and select the scored point, the starting point included, whose z500 RMSE averaged over the daily lead times from 3 to 10 days is lowest. We do not stop early on the increments between neighboring checkpoints: on 49 initializations those increments are comparable to the run-to-run variation measured in Appendix H.6, so a threshold on them would select on noise. Each model is selected independently: the same rule, the same grid of scored points and the same 49 initializations, applied to its own curve. We do not transfer to WRAP-former-Spherical the number of updates selected for the frame lift. The two curves have different shapes, so a checkpoint chosen on one model's curve is not the other model's best scored checkpoint, and WRAP-former-Spherical is a row of Table 8 rather than a diagnostic. Selecting each model on its own curve gives both the same rule and the same selection multiplicity, as we already do for the step length (Appendix H.4). On the curves of Table 4 the two models happen to select the same number of updates, but that is an outcome of the rule and not an assumption of it. If the starting point is selected, the model reported as WRAP-former-Frame is the model after the first stage, and likewise for WRAP-former-Spherical.

Table 4: Every point the stage-2 selection sees: z500 RMSE ($m^2\,s^{-2}$) of the 6-hour rollout averaged over the daily lead times from 3 to 10 days, on the 49 initializations of 2019, for the stage-1 starting point and each scored milestone. Lower is better, and the selected point of each model is in bold. The two rows are the two lifts under stage-2 fine-tuning, and the selected checkpoint of each is the model that Table 8 reports as WRAP-former-Frame and as WRAP-former-Spherical. The whole grid is given so that the selection can be checked, and any other rule applied to it, without taking our word for the argmin. Selection uses 2019 only. No score on 2020, the year of every other table, enters it. *The 10,000-update checkpoint of WRAP-former-Frame is not one of the scored points of the grid and was not a candidate. We scored it afterwards to check that the optimum is not an artifact of the spacing of the grid. It is worse than 9,000 under this criterion and under averages over 3–5, 5–10 and 7–10 days, so the frame lift curve turns over at 9,000. A dash marks a point we did not score.

	Updates of stage 2				
Model	0 (stage-1 end)	4,000	7,000	9,000	10,000*
Frame lift	473.4	437.6	441.0	**418.6**	426.7
Spherical-coordinate lift	492.5	462.0	421.6	**421.5**	–

Stage 2: what the selection does and does not control. The criterion, the grid and the verification year were fixed before the curves were scored. The grid is $\{0, 4\text{k}, 7\text{k}, 9\text{k}\}$ because a checkpoint is scored only where the curriculum changes character, and scoring every thousandth checkpoint on 49 initializations would cost more than the fine-tuning itself. Selection is an argmin over the same

grid of four points for both models, and Table 4 reports all of it. Two limitations are worth stating. First, the criterion is a single scalar, and it weights the long lead times, where the errors are large, more than the short ones. The point it selects for WRAP-former-Frame is the argmin under every aggregation we checked, over 3 to 5, 5 to 10 and 7 to 10 days as well as 3 to 10. The point it selects for WRAP-former-Spherical is not: that curve is flat between 7,000 and 9,000 updates, 0.02% apart, so aggregating over 7 to 10 days would select 7,000 instead. We use one aggregation for both models rather than the one that suits each. Second, the milestones are scored in fp32 while the numbers we report on 2020 are produced in bf16, and the change of precision moves this criterion by up to 0.97%, which is of the same order as the gaps between neighboring milestones. None of these touches the comparison between the two models, which is made on one grid, one criterion and one precision.

Rotated patches and exactness. Orientation 1 uses the resampling weights (15), estimated by binning 16×16 sub-pixels per pixel, and orientation 3 is orientation 1 relabeled by γ^2. At our resolution the half turn γ^2 of the pixel array misses a geometric half turn by 0.015 pixel spacings for the median pixel, while the quarter turn γ misses one by 0.26 and by up to 1.9, which is why we resample instead. Each entry of orientation 1 draws on one to eight pixels, so the resampling smooths the field. We rotate wind components in physical units before normalization and predicted increments without the mean offset, since an increment is a difference of two vectors. In float64 the frame lift with north-aligned frames is equivariant to within 10^{-8} under the quarter turns and the half turn, at initialization and at 24k, 48k and 96k steps.

Loss on every orientation. The loss of Appendix C.4 scores orientations 0 and 2 with the full objective (18), and orientations 1 and 3 with its grid term only, at the weight $1 - \lambda$, since the spectral term requires values at the pixel centers. Each orientation carries one quarter of the total weight. For the first stage of WRAP-former-Frame, the worse of orientations 0 and 2 stays within 1% of the output in MSE on the same four states at 24k, 48k and 96k steps.

Cost. We train mostly on NVIDIA B200 GPUs, with some segments on RTX PRO 6000 GPUs, and we compile most segments with `torch.compile`. We measure cost on compiled B200 segments only, with a fused attention kernel in which the zero key is one extra key column. Per training step, the frame lift costs 3.9–5.1× as much as the spherical-coordinate lift. We measure the ratio from the median optimizer-step times of the training runs themselves, with identical compilation, batch size and gradient accumulation. It falls from $5.1\times$ on one GPU to $3.9\times$ on eight, because the gradient all-reduce takes a larger share of the spherical-coordinate lift's step time, and on one and two GPUs it exceeds the $4\times$ token ratio because attention is quadratic in the number of tokens. Over the whole run, including uncompiled segments, segments on RTX PRO 6000 GPUs, job restarts, compilation, validation every $1{,}000$ steps and checkpoint writes, the frame lift has cost 4.7 GPU-seconds per step, 20–29% above its compiled step time. A run to the end of a cooldown started at 48k steps, i.e., 59,750 steps, therefore costs about 78 GPU-hours. The width controls match the frame lift in FLOPs but run far faster, since a wider matmul uses the GPU better than more tokens do. On one compiled B200, over 12 optimizer steps, the frame lift takes 3.78 seconds per step and axial RoPE takes 0.76 at width 1024 and 1.87 at width 2304, the nearest width we benchmarked. Scaling the quadratic part to width 2048 puts the width controls at about 0.4 of the frame lift per step. Stage 2 adds rollout updates whose cost per update grows with the curriculum from $K = 2$ to $K = 10$. At the same per-step cost and the curriculum's mean rollout length this is roughly 70 GPU-hours. The first stage of WRAP-former-Frame runs the full schedule of 288,000 steps, which at 4.7 GPU-seconds per step is about 376 GPU-hours, so WRAP-former-Frame costs on the order of 450 GPU-hours in total. The cooldown runs that the controlled comparison scores are far cheaper: about 78 GPU-hours for the frame lift row and, at the spherical-coordinate lift's 86 GPU-hours for $288{,}000$ steps, about 18 for each single-frame row.

H.3 From Stormer to our recipe

We start from Stormer (Nguyen et al., 2024) and change a component only when a symmetry requirement, a measured failure or a comparison calls for it. We keep its grid-independent choices, namely the interval sampling, the prediction target and the interval conditioning. We change the grid to HEALPix with one linear map per patch, add vertical velocity and 6-hour precipitation so

that the state closes under continuity and can be scored on SEEPS(Rodwell et al., 2010), supply $\sin(\text{latitude})$ so the half turn maps the Coriolis parameter correctly, form queries and keys without the interval conditioning and add a zero key to every head (Appendix H.13), add the spectral term to the objective (Appendix H.14), and replace the cosine schedule with a constant rate and a final cooldown (Hägele et al., 2024), which lets us score any checkpoint without retraining. For stage 2 we grow the rollout one step at a time as GraphCast and AIFS do, stopping at 10 steps so the longest rollout reaches the longest lead we score, at the learning rate of Stormer's first rollout stage. On the 49 initializations of 2019 the selected stage-2 checkpoint of WRAP-former-Frame lowers the z500 RMSE of the 6-hour rollout averaged over 3 to 10 days by 11.6% relative to the stage-1 cooldown it starts from. Appendix H.4 defines the inference conventions and Appendix H.5 measures them.

H.4 Inference conventions

Stormer averages forecasts along several sequences of steps, which lowers RMSE in the same way that averaging ensemble members does, so its published scores are not comparable with a single deterministic forecast. We therefore report the two conventions below and compare each with baselines of the same kind.

A model conditioned on the forecast interval can reach a lead time T along any sequence of steps from $\{6, 12, 24\}$ h that sums to T. We use this freedom in two ways and keep every choice fixed across initializations and variables.

Deterministic forecast. We roll the model out along a single sequence of steps of one length. For WRAP-former-Frame and WRAP-former-Spherical, we choose between 6-hour and 24-hour steps separately for each model on the 49 initializations of 2019, by the z500 RMSE averaged over the daily lead times from 3 to 10 days. Both select 24-hour steps, though for WRAP-former-Spherical the two lengths are 0.1% apart on 2019, inside the run-to-run variation, so that choice is not resolved. Table 5 reports both step lengths for WRAP-former-Frame on 2020. The controlled comparison and the ablations use 6-hour steps for every model, and precipitation is always scored from the 6-hour rollout, whose 24-hour accumulation is the sum of four steps. Unlike Stormer's reported protocol, this forecast is one rollout and is therefore comparable with the deterministic models of Table 8.

Mean over step sequences. For each lead time T we build K step sequences by a fixed rule and average their forecasts. We enumerate the multisets of $\{6, 12, 24\}$ h that sum to T, shortest first and ties broken lexicographically, and take from each its ascending ordering, then further orderings drawn at random with a fixed seed shared by all initializations, until we hold K distinct sequences or the multisets run out, e.g., the six that reach 24 hours. Unlike Stormer, which selects the best 32 of 128 random sequences on validation data, we select no sequence on data. Sequences share prefixes, so we compute every distinct intermediate state once, which for 10 daily lead times with $K = 8$ takes 153 forward passes per initialization instead of 40 for a single chain of 6-hour steps.

Probabilistic ensemble. We perturb the initial state once per member and roll out in 6-hour steps, with $M = 50$ members, the size of IFS ENS and NeuralGCM ENS. We compare two perturbations. The first is white Gaussian noise on the pixels, each channel at the amplitude the second draw gives it, so the two differ in spatial structure alone. Stormer perturbs the initial state with Gaussian noise but states neither its amplitude nor which fields carry it, so this is our construction of that idea rather than a reproduction of theirs. The second draws every channel in spherical-harmonic space up to degree 64 with a per-degree amplitude of α times the square root of the climatological power of that channel's 6-hour increment at that degree, so each field is perturbed in proportion to how much it changes over 6 hours at every scale. For both we set the perturbation fraction α on the 49 initializations of 2019 so that the spread of the ensemble matches the error of its mean, the consistency relation of Leutbecher & Palmer (2008), minimizing the mean of $(\ln \mathrm{SSR}_c)^2$ over z500 and T850 and the ten daily lead times (Appendix H.11). The criterion weights over- and under-dispersion alike, so α is not tuned to the ratio we report, and we do not choose it by CRPS, since a larger perturbation also degrades every member. Each perturbation therefore enters at its own calibrated size, and the comparison is of spatial structure rather than of amplitude. The sizes that result, α of 1.03 and 0.72, are of the order of a whole 6-hour increment and so far larger than analysis error: α is what disperses a deterministic model, and we do not read it as a representation of initial-condition uncertainty. Neither ensemble carries a model-error term: the members share

one set of weights and one step length, so they sample initial-condition uncertainty alone, and an ensemble built that way is expected to remain under-dispersed at long lead (Leutbecher & Palmer, 2008).

Stormer's own ensemble, in its Appendix C.4, perturbs the initial state the same way, with $M = 128$ members formed as four noise draws on each of its 32 selected step sequences, so it combines both conventions above where ours is a perturbation ensemble alone. It reports the result graphically rather than numerically, so we compare protocols rather than scores. Table 5 reproduces its finding that perturbing the initial state improves the probabilistic metrics and costs deterministic skill: our ensemble mean is 360 at 5-day z500 against 270 for the mean over $K = 8$ step sequences. We therefore do not offer this ensemble as a competitive probabilistic forecast, and it is absent from Table 8: its amplitude is set for calibration rather than for skill, and the ensemble means listed there average a forecast that is already step-sequence averaged. What the comparison isolates is the perturbation's spatial structure at a matched amplitude, nothing more.

H.5 Inference-time ablations

Both conventions of Appendix H.4 act at inference only, so we can measure each one on a single trained model. Table 5 starts from a deterministic rollout of WRAP-former-Frame and varies the step length, then the number of averaged step sequences, with everything else fixed. Averaging lowers RMSE partly by smoothing, so we report the CRPS beside it, which for a single forecast is the mean absolute error.

Shaping the perturbation helps more the longer the forecast runs, and costs skill at the start. Against the white-noise rung at its own calibrated amplitude, the shaped draw lowers z500 CRPS by 4.0% at 5 days and 10.5% at 10, and T850 CRPS by 4.7% and 5.8%, but raises both at 1 day, by 5.7% on z500 and 11.4% on T850, and is level at 3 days on z500. A draw carrying the increment's own spectrum projects onto the scales that grow, so its benefit accumulates with lead, while a flat draw puts most of its variance where the model damps it within hours. Calibration follows the same pattern: the corrected spread–skill ratio is nearer one at every lead on z500 and from 3 days on T850, where at 1 day the shaped draw overshoots to 1.29 against white noise's 0.89. The rank histogram agrees on T850, where the mass in the end bins falls from 12.8 to 8.9% at 5 days, and disagrees on z500, where the last bin rises from 5.8 to 8.5 times the flat value while the first falls from 0.92 to 0.56. That asymmetry is one-sided, so it is a conditional bias rather than a failure of dispersion, and the wider white-noise spread was masking it.

Table 5: Inference-time choices for the trained WRAP-former-Frame, varied one at a time (2020, WeatherBench 2 protocol, all 732 initializations). z500 RMSE and CRPS are in $\mathrm{m^2\,s^{-2}}$ and T850 RMSE in K. "Passes": forward passes per initialization for 10 daily lead times, summed over the members of an ensemble. α: the perturbation fraction of the row, chosen on 2019 so that spread matches error (Appendix H.4). For a single forecast, deterministic or a mean over step sequences, the CRPS equals the mean absolute error and SSR does not apply. For an ensemble, RMSE is that of the ensemble mean. A calibrated ensemble of $M = 50$ has $\mathrm{SSR} = 0.99$ and a lower value is under-dispersed. Lower is better for RMSE and CRPS, and "–" marks a quantity that does not apply.

			RMSE z500			RMSE T850		CRPS z500		SSR z500
Inference	Passes	α	3 d	5 d	10 d	5 d	10 d	5 d	10 d	5 d
Deterministic, 6-hour steps	40	–	129	281	725	1.60	3.30	163.8	429.4	–
Deterministic, 24-hour steps	10	–	133	284	663	1.59	2.97	165.4	396.4	–
Mean over $K = 2$ step sequences	21	–	127	274	644	1.54	2.90	160.1	386.1	–
Mean over $K = 4$ step sequences	82	–	128	274	648	1.54	2.92	159.8	388.6	–
Mean over $K = 8$ step sequences	153	–	125	270	649	1.52	2.92	157.7	389.0	–
Ensemble, white Gaussian noise on the initial state	2,000	1.03	195	360	698	1.92	3.05	179.1	359.8	0.81
+ per-degree amplitude from the 6-hour increment climatology	2,000	0.72	200	360	677	1.88	2.98	171.9	322.0	0.94

H.6 Controlled comparison

Table 6 isolates the treatment of symmetry: every row shares data, schedule and depth, and the rows differ as Section 5.3 states, with two rows at twice the width. The objective is shared up to the one difference the caption states, since the spectral term of (18) is not defined at the resampled positions of two of the frame lift's four orientations. The one row not matched in steps is axial RoPE with $\mathrm{SO}(3)$ augmentation, which was halted (Appendix H.14).

The shortest leads. The bottom panel of Figure 5 plots every 6-hourly lead; the top starts at 1 day. The points before 1 day carry more initialization noise than the rest, and they are also the points most favorable to the frame lift, which reaches -10.5% at 6 hours against -5.8% at 1 day. We show them because the benchmark reports the full lead range, and we read the margin at and beyond 1 day as the resolved one.

Equal training compute. Matching the cost of the frame lift per step forces a change of tokenization or width, so we also compare at equal *total* compute, where nothing structural changes. Stage 1 of the frame lift to the end of its cooldown from 48k steps costs about 78 GPU-hours, and the spherical-coordinate lift trains its whole 288k schedule for 86, measured over every segment of both runs. At that budget the frame lift has the lower RMSE at every lead time we score, by 2.8% at 1 and 3 days, 1.7% at 5 and 0.9% at 10 for z500, and by 1.9% to 2.3% for T850. Two asymmetries work against it: the spherical-coordinate lift sees 9.2M samples against 1.9M for those GPU-hours, and the frame lift trains with half its spectral weight. The spherical-coordinate lift is also not compute-starved, since its own cooldowns from 48k, 144k and 288k steps score 315.3, 316.6 and 312.0 at 5-day z500, i.e., six times the training buys it about 1%. Width behaves the same way. At twice the width and 32 heads, which matches the frame lift in linear-layer FLOPs, the spherical-coordinate lift scores 317.5 at 5-day z500 against 315.3 at its own width, and axial RoPE moves from 316.6 to 317.0; both differences are inside the run-to-run variation below. This comparison holds the tokenization fixed and asks whether the lift earns its cost. It does not separate the exact group from the token budget, which no experiment here does.

Trend with training. The rows are scored after a cooldown from 48k steps, a sixth of the schedule of Table 8, and the margins narrow over the one doubling we can compare. On the 49 initializations of 2019 used for selection, between the cooldown from 24k steps and the cooldown from 48k, the frame lift's lead in 5-day z500 falls from 6.0% to 3.1% over the spherical-coordinate lift, from 4.5% to 0.7% over axial RoPE and from 1.5% to 0.9% over learned embeddings. One doubling does not extrapolate to the full schedule, and we do not claim that these differences survive it.

Run-to-run variation. Every row is a single training run. To measure how much two runs differ by chance, we compare two runs of the spherical-coordinate lift without value rotation in the earlier configuration of Appendix H.13 whose configurations are identical. They differ in random quarter turns about the polar axis applied to the training samples and in how the batch of 32 was split into gradient-accumulation steps, neither of which changes a training step in exact arithmetic, since the model is exactly equivariant to these turns and the objective is invariant under them. They also saw different sample orders, since each was resumed from checkpoints saved within an epoch and a resume restarts the epoch's sequence. After the cooldowns started at 24k and 48k steps, their RMSE on the 49 initializations of 2019 differs by $+0.36\%$ and -0.14% averaged over 18 variable–lead cells, by up to 1.0% averaged over the cells of one lead time, and by up to 1.9% in a single cell. As the two runs share their random seed, this is a lower bound on the variation between independent runs, and we treat differences between rows below about 1% as unresolved.

H.7 Ablations

Table 7 varies value rotation in the frame lift, adds the equatorial reflection to the frame lift as augmentation, varies the frames the frame lift assigns to the four tokens of a patch, and varies value rotation in the spherical-coordinate lift, under the protocol of Table 6. The frames row asks what a common north is worth. Appendix B.2 defines the two sections and shows that they agree on the exact group, the token count and the cost, and differ only in whether every token's frame shares a common north: block frames take each patch's frame from the patch, which is all the lift itself provides. Block frames also turn the two horizontal wind components into each other, so these runs use the shared wind normalization of Appendix H.2 (*Input normalization*), which is inert for a north-aligned lift. Block frames are worse on every variable and lead of the table. Against the north-aligned frame lift at the matched cooldown from 24k steps, scored on the 49 initializations of 2019 that the paper selects on, they raise z500 by 2.4% at 3 days and 3.2% at 5, and T850 by 2.2% and 2.3%, all larger than the run-to-run variation. We compare the two arms there rather than on 2020 because both were scored under one diagnostic protocol at that cooldown, whereas the table's other rows are cooldowns from 48k steps. The common north is therefore worth about three percent,

Table 6: Controlled comparison: latitude-weighted RMSE (z500 in $\mathrm{m}^2\,\mathrm{s}^{-2}$, T850 in K) of deterministic 6-hour rollouts over all 732 initializations of 2020, each scored after the cooldown from its 48k-step checkpoint except the row marked ♯. All rows share data, schedule and depth, every model with Wigner-RoPE uses value rotation, and lower is better throughout. "Aug.": random rotations from the stated group; "–" in "Aug." and "Exact group" means none. "Width 2048": twice the width and 32 heads at half the learning rate, reported for axial RoPE and for the spherical-coordinate lift so that capacity against orientations is asked both outside and inside the equivariant family. "Cost": linear-layer FLOPs per training step relative to the spherical-coordinate lift, at which the two width controls and the frame lift are matched. The objective is not shared exactly: the frame lift carries half the spectral weight of the other rows and half its orientations are supervised against resampled targets, both of which work against it (Appendix C.4). Most differences narrow between 5 and 10 days and the ordering of the two lifts is unchanged, while the ordering of the three augmented axial rows is not resolved. ♯Collapsed at 22.4k steps and halted at 34.5k, scored after the cooldown from 24k, and not scored at 10 days (Appendix H.14).

			z500			T850		
Position	Aug.	Exact group	3 d	5 d	10 d	3 d	5 d	Cost
Learned embedding	–	–	139.9	315.0	825.3	1.091	1.788	1.0
Axial RoPE	–	–	140.7	316.6	822.7	1.093	1.793	1.0
Axial RoPE	C_4	–	141.9	319.8	823.3	1.097	1.804	1.0
Axial RoPE	D_4	–	143.3	320.8	825.4	1.107	1.814	1.0
Axial RoPE	$\mathrm{O}(2)$	–	145.5	323.9	824.9	1.120	1.824	1.0
Axial RoPE	$\mathrm{SO}(3)^\sharp$	–	159.4	346.1	–	1.184	1.912	1.0
Axial RoPE, width 2048	–	–	141.5	317.0	821.9	1.092	1.794	4.0
Spherical-coordinate lift, width 2048	–	C_4	141.7	317.5	822.1	1.097	1.793	4.0
Spherical-coordinate lift	–	C_4	141.3	315.3	816.7	1.097	1.791	1.0
Spherical-coordinate lift	D_4	C_4	144.4	322.6	821.2	1.111	1.819	1.0
Frame lift	–	D_4	133.9	306.7	809.8	1.058	1.744	4.0

even though neither choice changes the group the model is exactly equivariant to, and the sign is the one a rotating planet suggests: its equations single out the same axis (Appendix H.1), so the common north is a prior and not bookkeeping. We verify numerically that the block-framed model is still equivariant to the quarter turn and to F to within 10^{-9}, so the penalty is a property of the prior and not of a lost symmetry.

Value rotation lowers the error of the frame lift at every lead time of the table and raises that of the spherical-coordinate lift, by around a percent in either direction, which is at the resolution of the run-to-run variation of Appendix H.6. The two lifts are different models, so the opposite sign is not a contradiction. Only the frame lift holds one patch in four frames at once, and without value rotation such a head keeps only the part of its values that quarter turns leave unchanged (Proposition 8). For a single-frame lift, Proposition 7 bounds the effect of transport by the frame mismatch between patches, which is small. The pattern follows that argument on the variable it concerns. For the frame lift, the steadiest gain is on the 850 hPa wind, 1.0% at both leads, the only genuinely vector-valued field in the table, while the two scalars range from 0.6% to 1.2%. For the spherical-coordinate lift, whose patches carry a single frame and which therefore has no cross-frame vector sum to correct, the wind is where value rotation matters least, 0.3–0.4% in the other direction. Augmenting the frame lift with equatorial reflections lowers its error further, by more than a percent at 5 days on both scalars, which is just outside that resolution and rests on one run. The equatorial reflection is improper, so no group of rotations contains it and neither lift makes it exact. We report it as an ablation, and the models of Table 8 do not use it.

H.8 Comparison with published systems

Table 8 compares WRAP-former-Frame and WRAP-former-Spherical with published systems. Section 5.3 and Figure 5 summarize it.

H.9 Additional scores and figures

Figure 7 shows one forecast of each two-stage model against ERA5, Figure 8 gives error against pressure level, Figure 10 maps where WRAP-former-Frame beats WRAP-former-Spherical, and Ta-

Table 7: Ablations: RMSE as in Table 6 (for the wind, of the vector, in $\mathrm{m\,s^{-1}}$), same protocol. "North": frames aligned with local north; "block": frames aligned with the axes of each patch's HEALPix base face instead. "reflection aug.": trained with random equatorial reflections, the improper symmetry of the rotating-sphere equations that the exact group does not contain. "Value rotation": with (✓) or without (–) value rotation (11). Lower is better in every column. The first row and the fifth are the frame lift and the spherical-coordinate lift of Table 6, repeated here as the reference of their block, and every other row changes one thing relative to the reference of its own block. [b]Scored after the cooldown from 24k steps rather than 48k, since this run was halted before 48k, so it is not matched to the other rows and should not be read against them. Matched against the frame lift with north-aligned frames at the same 24k cooldown, on the 49 initializations of 2019, block frames raise 5-day z500 by 3.2% and 5-day T850 by 2.3%, which is resolved against the run-to-run variation of Appendix H.6.

			z500		T850		Wind 850 hPa	
Model	Value rotation	Frames	3 d	5 d	3 d	5 d	3 d	5 d
Frame lift	✓	north	133.9	306.7	1.058	1.744	3.03	4.96
	–	north	135.5	308.6	1.066	1.760	3.06	5.01
	✓	north, reflection aug.	132.6	302.4	1.051	1.726	3.01	4.92
	✓	block[b]	143.1	322.0	1.100	1.809	3.15	5.12
Spherical-coordinate lift	✓	north	141.3	315.3	1.097	1.791	3.11	5.05
	–	north	140.0	313.4	1.090	1.781	3.10	5.03

ble 9 gives precipitation and wind. Precipitation is scored as the model predicts it, negative values included, which is how WeatherBench 2 scores GraphCast, and a 24-hour accumulation is the sum of four consecutive 6-hour steps, so Table 9 scores our models from a 6-hour rollout. Table 10 gives the remaining variables of the evaluation protocol and the anomaly correlation, and Table 11 the physical consistency of the forecasts.

Against the non-equivariant baseline the same comparison is one-sided. At the matched 48,000-step checkpoints, where the two differ only in how position enters attention, the frame lift is ahead in every latitude band: by 5.4% in the Arctic (poleward of 60°N), 5.2% in the tropics (within 20° of the equator) and 4.2% in the northern mid-latitudes, and by least, 1.0–1.1%, in the southern mid-latitudes and the Southern Ocean storm track. No band favors the baseline. The storm track is where the absolute error is largest, roughly $545\,\mathrm{m^2\,s^{-2}}$ there against 303 in the northern mid-latitudes, and it is the region whose flow is most strongly dominated by fast baroclinic growth, where an architectural prior has least purchase.

Precipitation. The SEEPS of our two-stage models is worse than GraphCast's at every lead time, and the cause is the objective rather than the encoding. Precipitation is intermittent, non-negative and heavy-tailed, and a squared-error objective is minimized by the conditional mean, which places light precipitation where the truth is either dry or heavy, and that placement is what SEEPS scores. The effect is general to the class: GraphCast scores 1.5 times worse than IFS HRES on SEEPS at 3 days while being ahead of it in every RMSE column of Table 8. It is not specific to the lift either, since WRAP-former-Spherical and WRAP-former-Frame differ by less than 0.005 at 1 and 5 days. The stage-1 and stage-2 rows of Table 9 localize it: the rollout curriculum that lowers the wind error at 3 days and beyond raises SEEPS at every lead, so categorical skill is traded for squared error. The gap from WRAP-former-Frame to GraphCast closes with lead time, from 15% at 3 days to 3% at 10. A strictly generative ensemble avoids the trade, since sharp members are not penalized when only the ensemble is scored, whereas a deterministic forecast verified pointwise cannot be both sharp and accurate. We are deterministic and inherit that pitfall. On the other axis of physical realism that WeatherBench 2 targets we do deliver: at 5 days our forecasts hold 0.97 and 0.93 of the power of ERA5 at planetary and synoptic scales, though only 0.47 at the mesoscale (Table 11), and at matched training the frame lift accumulates less grid-scale noise than the non-equivariant baseline (Figure 9).

H.10 Comparability of published results

What the comparison does not control. Three conditions work against our rows, and we state them rather than adjust for them. First, we train at 1.5°, whereas GraphCast, Pangu-Weather and FuXi train at 0.25° and are scored at 1.5°, so they see 36 times as many grid points per state while

Table 8: Skill on 2020 under the WeatherBench 2 protocol ($1.5°$, verified against ERA5): latitude-weighted RMSE, z500 in $m^2\ s^{-2}$, T850 and T2m in K. Lower is better in every column. All our rows use all 732 initializations of 2020. The official FuXi and Keisler files are incomplete, so those rows average over 702 and over as few as 727 initializations (Appendix H.10). †: official WeatherBench 2 scores. ‡: authors' values for 2020 at $1.5°$ against ERA5 (Appendix H.10). §: trained on years after 2020. ‖: official WeatherBench 2 forecasts, scored here by the same pipeline as our own rows, on the 178 initializations they provide (00 UTC every second day) rather than all 732. This model also runs at $1.4° \times 0.7°$, coarser meridionally than the others. Ensemble means and our mean over $K = 8$ step sequences (Appendix H.4) are listed separately, since averaging lowers RMSE at long leads. "–": variable not provided; "n/r": not reported numerically. Further variables and anomaly correlation: Appendix H.9.

	z500			T850			T2m		
Model	3 d	5 d	10 d	3 d	5 d	10 d	3 d	5 d	10 d
Climatology†	820	821	822	3.44	3.44	3.45	2.61	2.61	2.61
Persistence†	956	1050	1128	4.33	4.59	4.82	2.94	3.21	3.50
IFS HRES†	137	305	803	1.18	1.84	3.65	1.13	1.52	2.69
Sphere-native									
GraphCast† (Lam et al., 2023)	124	274	732	0.95	1.56	3.36	0.82	1.25	2.56
Keisler† (Keisler, 2022)	175	346	787	1.23	1.87	3.56	–	–	–
HEAL-ViT† (Ramavajjala, 2024)	118	271	763	0.96	1.60	3.49	0.85	1.27	2.57
WeatherMesh-4†§ (WindBorne Systems, 2025)	114	251	608	0.92	1.43	2.75	0.79	1.12	2.03
Spherical CNN‖ (Esteves et al., 2023)	161	338	816	1.18	1.86	3.63	–	–	–
Graph-FM‡ (Oskarsson et al., 2024)	n/r	363	825	n/r	n/r	n/r	n/r	1.57	2.82
Latitude–longitude									
Pangu-Weather† (Bi et al., 2023)	134	294	778	1.06	1.71	3.55	0.92	1.39	2.68
FuXi† (Chen et al., 2023)	125	277	632	0.98	1.60	2.92	0.85	1.27	2.17
ArchesWeather-Mx4† (Couairon et al., 2024)	132	289	694	1.01	1.63	3.12	0.83	1.28	2.32
Hybrid									
NeuralGCM† (Kochkov et al., 2024)	116	267	751	0.97	1.59	3.43	–	–	–
Ensemble means									
IFS ENS†	133	279	623	1.11	1.62	2.82	1.07	1.38	2.15
GenCast† (Price et al., 2025)	123	262	607	0.95	1.48	2.74	0.80	1.17	2.02
NeuralGCM ENS† (Kochkov et al., 2024)	126	266	607	1.03	1.54	2.76	–	–	–
ArchesWeatherGen† (Couairon et al., 2024)	130	272	610	0.98	1.52	2.75	0.83	1.21	2.05
Stormer† (Nguyen et al., 2024)	134	286	666	1.02	1.60	3.00	0.94	1.31	2.24
Graph-EFM‡ (Oskarsson et al., 2024)	n/r	399	695	n/r	n/r	n/r	n/r	1.64	2.32
MOSAIC‡ (Zhdanov et al., 2026)	n/r	n/r	624	n/r	n/r	2.78	n/r	n/r	2.05
Ours: WRAP*-former, deterministic*									
Spherical	139	295	687	1.05	1.64	3.07	1.02	1.39	2.34
Frame	133	284	663	1.02	1.59	2.97	1.00	1.35	2.26
Ours: WRAP*-former, mean over* $K = 8$ *step sequences*									
Frame	125	270	649	0.98	1.52	2.92	0.98	1.31	2.23

training as we do. Second, the frame lift is constrained to be exactly equivariant to D_4, which removes degrees of freedom an unconstrained model keeps, and costs 4.9–5.1 times as much per step as the spherical-coordinate lift on one and two GPUs (Appendix H.2, *Cost*). Third, the spectral term carries a third of the loss weight of the frame lift and half that of the spherical-coordinate lift, since the frame lift keeps the full grid weight $1 - \lambda$ but only $\lambda/2$ of the spectral weight (Appendix H.1). The two objectives share a minimizer, so this costs nothing asymptotically (Appendix H.14), but at the training lengths we reach it spends a third to a half of the gradient on degrees that carry little of the grid error, and grid error is what these tables report. That spending is what buys the spectral fidelity of Figure 9 and Table 11, so the RMSE columns understate the model. We do not claim an advantage in training length: our first stage sees 9.2M samples against the 9.6M of GraphCast's schedule, so the two are matched on that axis.

Kinds of baseline. The weather benchmark has two kinds of baseline. *Controlled baselines* share our backbone, data, objective and number of training steps and differ only in the treatment of symmetry, at equal depth and with parameter counts within 1% of the spherical-coordinate lift, except

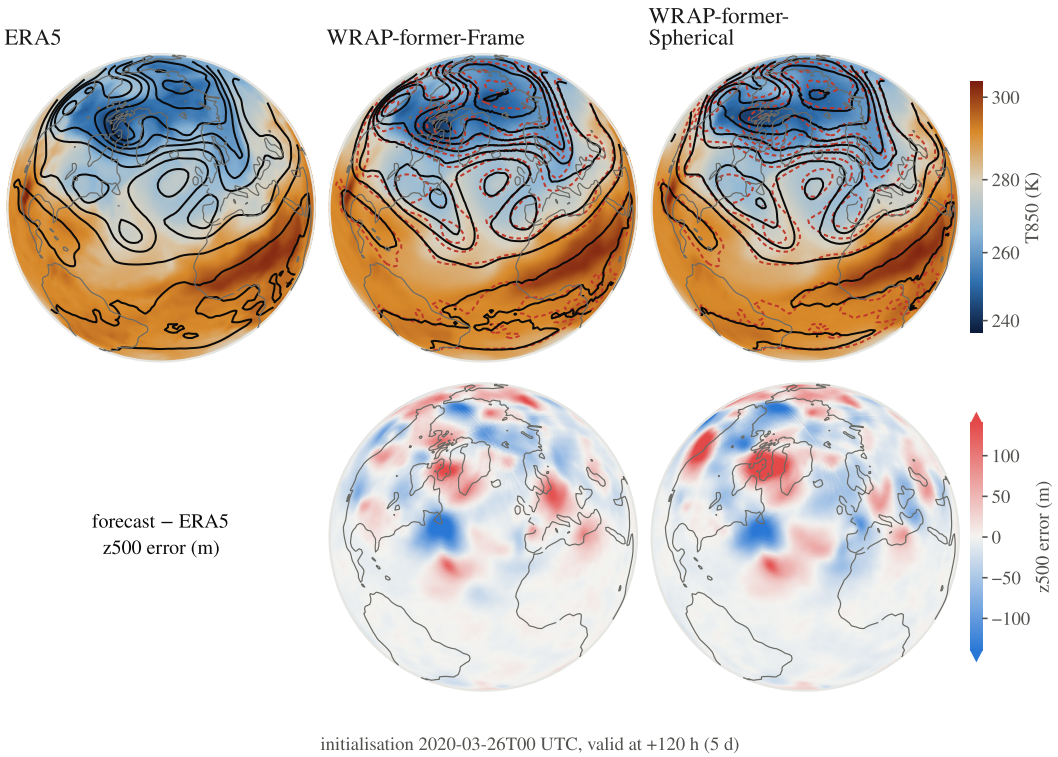


Figure 7: One forecast of the two models of Table 8, both at their selected two-stage checkpoints and both from the same initialization, 2020-03-26 at $+120$ h, the day at which axial RoPE's median 5-day z500 error falls, so neither of our models chose it. Top: 850 hPa temperature with 500 hPa geopotential contours, the verifying ERA5 contour repeated dashed on each forecast, so where the solid line leaves the dashed one is the error. Bottom: the z500 error of each. The two reproduce the same synoptic pattern and differ in where their errors sit: here WRAP-former-Frame errs by 29.0 m against 34.1 m for WRAP-former-Spherical, and the two forecasts differ by 28.3 m, as large as either error. A single initialization illustrates the task rather than establishing skill. The difference over the year is Figure 10 and Table 8.

the two width-2048 controls and axial RoPE with $\mathrm{SO}(3)$ augmentation, which was halted and is scored after the cooldown from 24k steps. The frame lift changes the cost of attention and of the linear layers without changing the parameter count, and matching the compute of the other rows would require a narrower frame lift or fewer steps, which would confound symmetry with model size, so we report its cost instead. The two width-2048 controls spend that extra compute at the tokenization every other row uses, which asks whether capacity substitutes for orientations outside and inside the equivariant family. *Published baselines* enter the tables only if they were scored on 2020 at $1.5°$ against ERA5, by WeatherBench 2 (†) or by their authors (‡).

Rows marked † are the official WeatherBench 2 results, computed by its maintainers. WeatherBench 2 lists HEAL-ViT (Ramavajjala, 2024) as Excarta (HEAL-ViT). Our implementation of RMSE and ACC reproduces these files exactly: scoring the official IFS HRES forecasts for 2020 with it matches every RMSE and ACC value of the published file for z500, T850, q700, T2m and MSLP at 1, 3, 5 and 10 days, in the global, tropical and extratropical regions. The same holds, to within 10^{-15}, for the bias and prediction activity of z500, the RMSE of the wind vector at 850 hPa and at 10 m in the global and tropical regions, and the SEEPS of 24-hour precipitation in the global, tropical and both extratropical regions. The fair CRPS of Ferro (2014), the RMSE of the ensemble mean and the spread–skill ratio match the official NeuralGCM ENS file for z500 at 5 days, over all 732 initializations and 50 members, to within 10^{-8}. These rows are scored exactly as ours. They differ in the model and its training data, and several of these models train on $0.25°$ ERA5. Our model forecasts on HEALPix, and we map its forecasts to the $1.5°$ grid by first-order conservative regridding (Appendix H.1), the operator by which WeatherBench 2 brings forecasts on other grids to $1.5°$. On 13 initializations of 2019 we compared it with bilinear interpolation and with the regular-

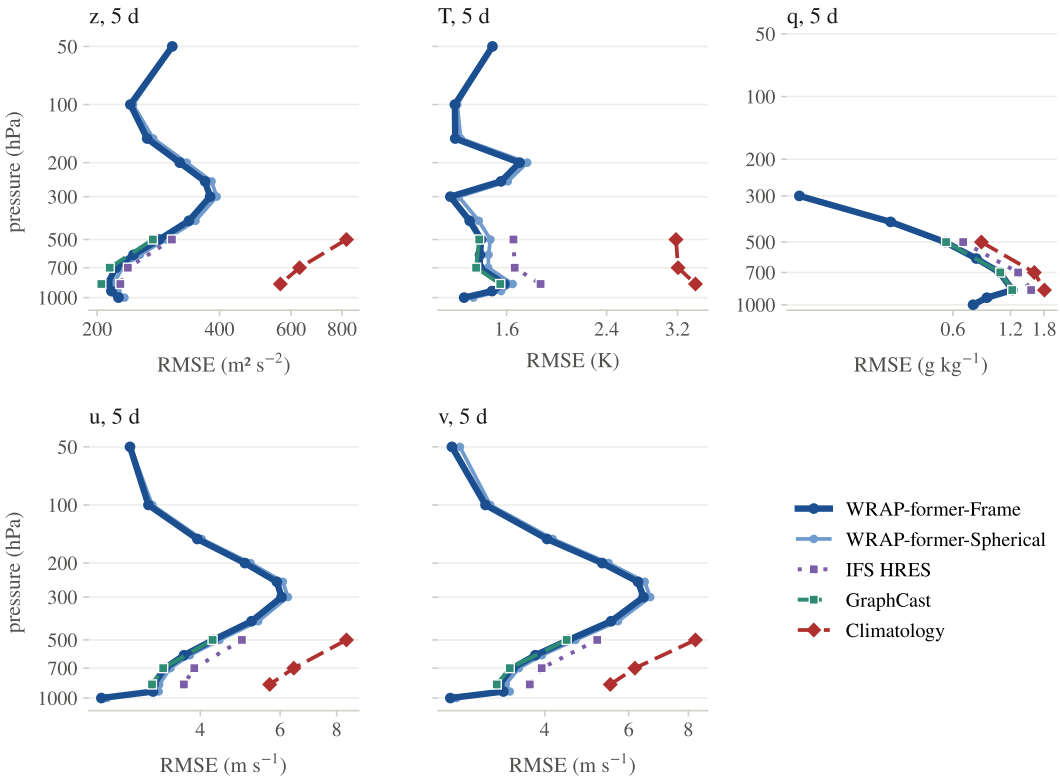


Figure 8: Error against pressure level at 5 days, 2020, for WRAP-former-Frame and WRAP-former-Spherical. Both are below IFS HRES at every level of every field, and they track GraphCast at the three levels we have access to, especially WRAP-former-Frame, while training at $1.5°$ against its $0.25°$. The advantage of the frame lift over the spherical-coordinate lift is flat with height rather than concentrated anywhere: 3.4 to 3.8% on geopotential at every level from 1000 to 200 hPa, and falling to nothing only at 50 hPa, where there is little error to remove. The error axis is logarithmic, because climatology is such an outlier. Specific humidity is plotted below 300 hPa only, since the stratosphere is dry and no model's error there is distinguishable from zero. IFS HRES, GraphCast and the climatology appear at 500, 700 and 850 hPa only, the sole levels their official WeatherBench 2 files store.

ized least-squares inverse of the map to HEALPix. Neither lowers our error at every lead time, so we keep the operator WeatherBench 2 itself uses. Rows marked ‡ are copied from the authors' papers, which score 2020 at $1.5°$ against ERA5 (Graph-FM and Graph-EFM initialize at 00 and 12 UTC each day of 2020, following WeatherBench 2, as we do): Graph-FM/Graph-EFM (Oskarsson et al., 2024), U-Cast (Rühling Cachay et al., 2026) and MOSAIC (Zhdanov et al., 2026). WeatherMesh-4 (WindBorne Systems, 2025) trains on 1971–2019 and 2021–2023 and has therefore seen years after the test year (§). Stormer's entry averages forecasts over step sequences and is listed with the ensemble means. The official FuXi forecasts end with the initialization of 16 December 2020 at 12 UTC, so its rows average over 702 of the 732 initializations. The official Keisler forecasts are missing between 2 and 5 initializations, depending on the variable and the lead time, so its rows average over the forecasts available at each lead time: 727 of the 732 for z500 at 3 days. WeatherBench 2 publishes 2020 forecasts for the spherical CNN of Esteves et al. (2023) but no precomputed scores, so we score them with the pipeline that produces our own rows. They cover 178 initializations, 00 UTC every second day from 1 January to 20 December 2020, and WeatherBench 2 documents no reason for the reduced sample.

Several relevant models have no comparable scores and are therefore absent: SFNO (Bonev et al., 2023), FourCastNet (Pathak et al., 2022) and FourCastNet 3 (Bonev et al., 2025a) report other years or $0.25°$ largely in figures, SMgNO (Hu et al., 2025) is evaluated on WeatherBench 1, Aurora (Bodnar et al., 2025) is scored only for 2022 and from operational initial conditions, DLWP-HEALPix (Karlbauer et al., 2024) reports 2017–2018, PEAR (Linander et al., 2025) trains on a reduced sub-

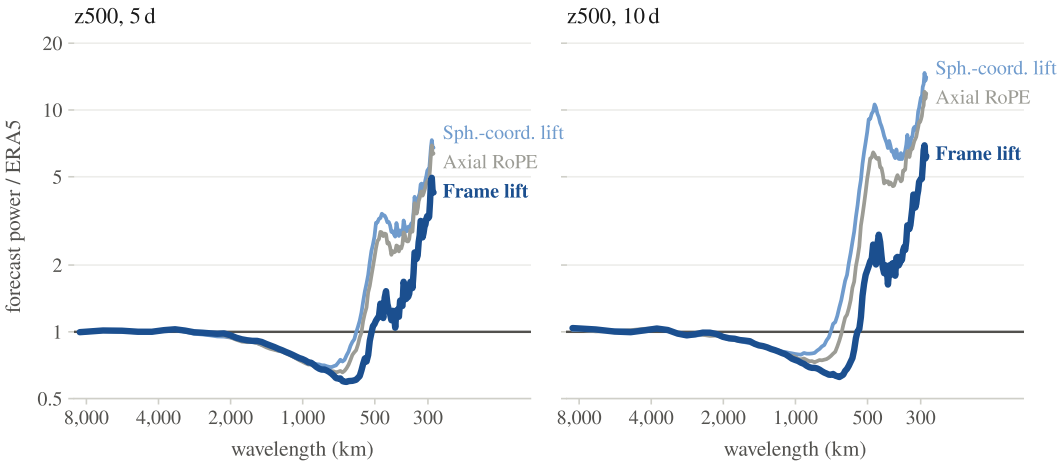


Figure 9: What a rollout does to each scale: the power of the forecast divided by ERA5's own power, at the matched 48,000-step checkpoints of the controlled comparison, so the three differ in how position enters attention and, for the frame lift, in the spectral weight its objective carries (Table 6). One is ERA5's power. Below it the forecast is damped, above it the forecast carries structure that is not in the atmosphere. At every resolved scale the three agree, within 0.99–1.04 from 8,000 down to about 1,000 km, and they differ only near the grid scale, where an autoregressive rollout accumulates spurious power: at 10 days and 400 km the frame lift reaches 2.0× ERA5 against 4.7× for axial RoPE and 6.9× for the spherical-coordinate lift. Our curve is also the lowest just above the grid scale, that is, slightly more damped there. As in WeatherBench 2 the power is computed along each latitude circle and averaged over $30° < |\text{lat}| < 60°$ at common wavelengths (Rasp et al., 2024).

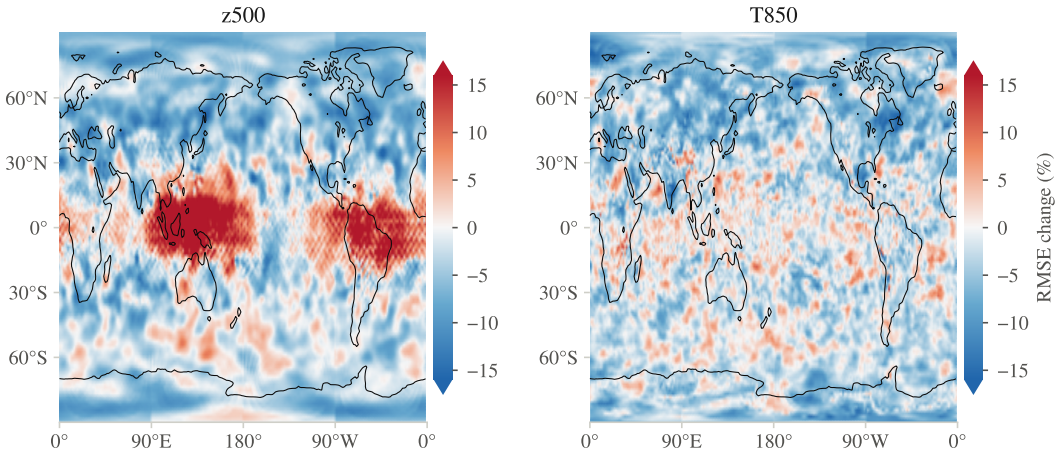


Figure 10: Where WRAP-former-Frame beats WRAP-former-Spherical, as the *relative* change in 5-day RMSE, 2020 (all 732 initializations). Per cent rather than absolute units, so the two panels share one scale. The advantage is not uniform: it *inverts* in a tight band about the equator, the red belt about the equator on the left. On z500 WRAP-former-Frame leads by 6.2% at 30–50°N and loses by 9.6% within 5° of the equator, the band where the Coriolis parameter passes through zero, so the flow is not rotationally dominated and $\sin(\text{latitude})$, the symmetry-breaking input, changes sign. The benefit of the encoding tracks how well the symmetry it builds in actually holds. On T850, whose error is not set by geostrophic balance, no inversion appears. Globally WRAP-former-Frame is ahead by 3.5% on z500 and 3.3% on T850, over 63% and 71% of the area.

set, and Spherical DYffusion (Rühling Cachay et al., 2024) emulates a climate model. For GenCast (Price et al., 2025) we use its official WeatherBench 2 scores for 2020.

Table 9: Precipitation and wind, 2020, under the protocol of Table 8 ($1.5°$, verified against ERA5, all 732 initializations for our rows): SEEPS, the stable equitable error in probability space of WeatherBench 2, for 24-hour precipitation, and latitude-weighted RMSE of the 850 hPa wind vector in $\mathrm{m\,s^{-1}}$. Lower is better in both column groups, which are different scores on different quantities and are not comparable with each other. Our rows are deterministic rollouts in 6-hour steps, which need not be the step length of our deterministic rows in Table 8 (Appendix H.4), and our precipitation is scored as the model predicts it, negative values included. "Stage 1" is the model before the second-stage rollout fine-tuning and the WRAP-former rows are after it: the curriculum lowers the wind error at 3 days and beyond and raises SEEPS at every lead, so squared-error skill is bought at the cost of categorical precipitation skill. For the frame lift the two SEEPS scores are 33% apart at 3 days and 4% apart at 10. †: official WeatherBench 2 score. "–": variable not provided.

	SEEPS, 24 h precipitation				Wind vector 850 hPa			
Model	1 d	3 d	5 d	10 d	1 d	3 d	5 d	10 d
IFS HRES†	0.205	0.331	0.466	0.761	1.85	3.33	5.23	9.19
GraphCast†	0.211	0.509	0.606	0.815	1.43	2.77	4.46	8.20
Pangu-Weather†	–	–	–	–	1.67	3.01	4.84	8.81
Spherical-coordinate lift (stage 1)	0.361	0.525	0.639	0.860	1.74	3.09	5.06	9.13
Frame lift (stage 1)	0.310	0.441	0.551	0.809	1.69	3.03	4.96	9.07
WRAP-former-Spherical	0.377	0.553	0.675	0.873	1.76	2.96	4.55	7.79
WRAP-former-Frame	0.375	0.587	0.671	0.843	1.71	2.90	4.51	8.14

Table 10: The three variables of the evaluation protocol that Table 8 does not carry, 2020, all 732 initializations for our rows: q700 in $\mathrm{g\,kg^{-1}}$, U10 in $\mathrm{m\,s^{-1}}$, MSLP in Pa, and the anomaly correlation of z500. Lower is better for RMSE and higher for ACC. Both lifts are ahead of IFS HRES and of GraphCast on 10-day ACC. †: official WeatherBench 2 score.

	q700			U10			MSLP			ACC z500	
Model	3 d	5 d	10 d	3 d	5 d	10 d	3 d	5 d	10 d	5 d	10 d
IFS HRES†	1.02	1.32	1.85	1.61	2.57	4.48	150	309	749	0.930	0.515
GraphCast†	0.80	1.06	1.59	1.32	2.18	4.01	132	276	682	0.943	0.574
WRAP-former-Spherical	0.84	1.07	1.47	1.48	2.28	3.71	150	298	640	0.934	0.600
WRAP-former-Frame	0.83	1.05	1.43	1.45	2.21	3.56	143	288	618	0.938	0.617

H.11 PROBABILISTIC METRICS

For an ensemble $\{x_m\}_{m=1}^{M}$ and verifying value y, we use the fair CRPS of Ferro (2014) and Leutbecher (2019),

$$\mathrm{CRPS} = \frac{1}{M}\sum_{m} |x_m - y| - \frac{1}{2M(M-1)}\sum_{m,m'} |x_m - x_{m'}|, \tag{19}$$

where M is the number of members, x_m and $x_{m'}$ are any two of them and y is the verifying value. It is an unbiased estimate of the CRPS of the distribution the members are drawn from at every $M \geq 2$, and it reduces to the absolute error for a deterministic forecast, which puts deterministic and ensemble forecasts on one scale. The spread–skill ratio (SSR) divides the ensemble spread, the square root of the unbiased variance of the members, by the RMSE of the ensemble mean, as in WeatherBench 2. A calibrated ensemble of M members has $\mathrm{SSR} = \sqrt{M/(M+1)}$, slightly below one, and a lower value is under-dispersed. Where a target independent of M is needed, as in the choice of the perturbation size, we use $\mathrm{SSR}_c = \sqrt{(M+1)/M}\,\mathrm{SSR}$, whose expectation is one at any M. All scores are latitude-weighted averages over grid points and initializations (Rasp et al., 2024).

H.12 PHYSICAL CONSISTENCY

Skill scores say how close a forecast is to the verifying state, not whether the forecast is a state the atmosphere could occupy. We therefore also measure how far the forecast drifts from identities

that the model is never taught, and whether it keeps the power of ERA5 at every scale. Table 11 reports the hydrostatic residual $\partial\Phi/\partial \ln p + R_{\text{dry}} T_v$ relative to the layer thickness and the residual of the continuity equation $\partial\omega/\partial p + \nabla_h \cdot \mathbf{V}$ relative to the divergence, both over every pair of adjacent pressure levels, where ω is the vertical velocity in pressure coordinates, $\mathbf{V}$ is the horizontal wind and $\nabla_h \cdot$ is the horizontal divergence on the sphere. Our state can express the second because it carries vertical velocity. We evaluate continuity in spherical-harmonic space, where the divergence of the winds is exact, over degrees 1 to 20, i.e., at wavelengths above about 2,000 km, the planetary and synoptic scales that winds on the 1.5° grid resolve. Including degrees up to 100 raises the residual of ERA5 itself from 47% to 74% of the divergence. The table also reports the ratio of forecast to ERA5 power in three scale bands, which detects a forecast that scores well by blurring. Neither identity is exact on ERA5 as distributed, so the verifying ERA5 states are the floor, and the comparison of interest is the excess over it. A forecast smoother than ERA5 can also fall below that floor, since the residuals are largest at small scales, so the residuals are read together with the power ratios. For scale, ERA5 vertical velocity paired with the winds of ten days earlier, a state with no balance at all, has a continuity residual of 139%, against 46.7% for ERA5 itself. Both of our models sit below that floor at 1 and 5 days and rise above it by 10 days, to 51.4% and 54.2%, so the continuity balance degrades over a rollout while the hydrostatic balance does not.

Table 11: Physical consistency against lead time, 2020, all 732 initializations. "Hydrostatic" and "continuity": root-mean-square residual of each identity relative to the term it balances, in percent, over all pairs of adjacent pressure levels, the second at degrees up to 20. ERA5 is the floor, and lower is better down to it. "Power ratio": forecast over ERA5 zonal power of 500 hPa geopotential at 5 days, over $30^\circ < |\text{lat}| < 60^\circ$, in the planetary, synoptic and mesoscale bands of Orlanski (1975), where 1 is neither blurred nor noisy.

	Hydrostatic			Continuity			Power ratio, 5 d		
Model	1 d	5 d	10 d	1 d	5 d	10 d	planetary	synoptic	mesoscale
ERA5 (floor)	0.57	0.57	0.57	46.7	46.7	46.7	1.00	1.00	1.00
WRAP-former-Spherical	0.54	0.54	0.54	44.0	43.6	51.4	0.96	0.94	0.53
WRAP-former-Frame	0.54	0.51	0.50	44.5	44.1	54.2	0.97	0.93	0.47

H.13 Training stability

Training on this task repeatedly failed in one way: the attention logits of a few heads grew without bound until the loss diverged, the instability described by Dehghani et al. (2023) and Wortsman et al. (2024). We identify two mechanisms that drove this growth in the first block and remove both. Earlier versions of our backbone instead clipped the query and key projections and capped the conditioning. Removing the mechanisms made those guards unnecessary, and none of them is active in any run we report.

Heads with nothing to attend to. Softmax weights lie on the simplex, so a head can switch off only by concentrating on one uninformative token, whose logit must keep rising to outcompete every other key. In an early run the query and key projections of four first-block heads became rank one and, at divergence, every first-block head placed all its mass on one distant token per query. The zero key, $k_\varnothing = 0$ with $v_\varnothing = 0$, has logit 0 for every query, contributes nothing to the output and is invariant since $\rho(R)\,0 = 0$, so a head with no informative key can rest on it at finite logits.

Conditioning inside the attention pattern. The forecast interval enters through adaptive layer normalization, $\hat{z}_i = \bar{z}_i \odot (1 + c_{\text{scale}}) + c_{\text{shift}}$, where $\odot$ is the elementwise product, $\hat{z}_i$ is the conditioned token, $\bar{z}_i$ is the normalized token i, and c_{scale} and c_{shift} are computed from the interval embedding. The shift, like the bias of the layer normalization, is shared by all tokens. A vector b shared by all tokens in the input to the query and key projections adds, among other terms, $(W_Q b)^\top \rho(R_i^{-1} R_j)(W_K b)$ to every logit, where W_Q and W_K are the query and key projections. This term is a content-free positional kernel whose sharpness grows with $\|b\|^2$. Under additive position embeddings, this term is constant and cancels in the softmax. Under rotary positions, it does not. Whenever a sharper positional pattern lowers the loss, the gradient therefore increases $\|b\|$, and because b is produced from one conditioning vector shared by every block, all heads sharpen

at once. In a run that already had the zero key, the adaptive shift grew eightfold and the bias of the layer normalization 2.3-fold, the first block's attention input grew sixteenfold, and the forecast collapsed within a few hundred steps. Forming queries and keys from the token normalized with a learned gain and no bias, without the conditioning scale or shift, gives the logits zero gradient into the conditioning vector.

The earlier configuration. Some runs predate the two changes above. That configuration applies the interval conditioning to queries and keys, has no zero key and no $\sin(\text{latitude})$ input, and augments with $\mathrm{SO}(3)$ by nearest-pixel permutation over 48 rotations without rotating the wind components. All four of its augmented runs failed under squared error, which is what motivated the changes.

Outcome. We log the gradient norm before clipping every 50 steps, and in no logged step of any run of the final configuration did it reach the clipping threshold of 1.0, so clipping is a backstop rather than what keeps these runs stable. The two changes do not remove every instability: late-block logit growth appears in the cooldowns of the frame lift without any rise in the gradient norm, and axial RoPE with $\mathrm{SO}(3)$ augmentation and axial RoPE at width 2048 were both halted, the latter retrained at half the learning rate as μP prescribes (Yang et al., 2021), which Table 6 scores.

H.14 Choice of objective

We compare the spectral objective ($\lambda = 0.5$) with plain squared error ($\lambda = 0$) on paired models that differ only in λ, in terms of skill, error spectra and training stability. The frame lift applies the spectral term only to the two orientations that predict at pixel centers, so its spectral weight is $\lambda/2$ while its grid weight stays at $1 - \lambda$ (Appendix H.1, *Objective*). Its pairs therefore contrast a spectral-to-grid ratio of $1 : 2$ against 0, where the other models contrast $1 : 1$ against 0. The uniform λ that matches this ratio is $1/3$, not 0.25, which would give $1 : 3$.

Theory. Every rotation-invariant quadratic loss on a scalar field on the sphere has the form $\sum_\ell w_\ell \sum_m |\hat{e}_{\ell m}|^2$, since the degree-$\ell$ subspaces are the inequivalent irreducible representations of $\mathrm{SO}(3)$, so by Schur's lemma an invariant quadratic form can only weight each degree as a whole. Grid squared error is the member $w_\ell = 1$ by Parseval, and the spectral term of (18) is the member $w_\ell = 1/(\sigma_\ell^2 + \epsilon)$, dropping the channel index. Within this family the weights change how the gradient is distributed across degrees but not the minimizer. For a degree with target amplitude a_ℓ, predicted amplitude $\hat{a}_\ell$ and coherence χ_ℓ between the two, the expected per-degree error is,

$$w_\ell\big(\hat{a}_\ell^2 + a_\ell^2 - 2\chi_\ell\, \hat{a}_\ell a_\ell\big), \qquad \hat{a}_\ell^* = \chi_\ell\, a_\ell, \tag{20}$$

where $\hat{a}_\ell^*$ is the amplitude that minimizes it, so every member of the family has the same minimizer, the conditional mean, which damps a degree by its predictability. The gradient with respect to the relative amplitude scales as $w_\ell \sigma_\ell^2$. Atmospheric increment power falls steeply with degree, so under grid squared error the small scales receive a small share of the gradient even where they are predictable. Whitening equalizes the learning pressure per unit of relative amplitude across degrees. The floor ϵ, at 5% of each channel's peak power, caps any degree's weight at $21\times$ that of the most energetic one. Two consequences follow. Whitening cannot make a forecast sharper than its predictability allows, since a degree with small χ_ℓ is damped to $\chi_\ell a_\ell$ whatever weight it carries. And the comparison against $\lambda = 0$ should show parity in grid RMSE rather than a win, since RMSE is what plain squared error optimizes, and after a cooldown we measure that parity, with median changes over 18 variable–lead cells of at most 1.1% on both 2019 and 2020.

The two effects are separable. The objective and the lift vary independently, and both arms of each exist at the cooldown from 48k steps, so the 2×2 is complete on 2020. At 5-day z500 the frame lift is ahead of the spherical-coordinate lift by 2.71% under the spectral objective, 306.7 against 315.3, and by 1.65% under plain squared error, 312.5 against 317.8. The advantage of the lift therefore does not come from the objective. It is smaller without the spectral term but still above the run-to-run variation of Appendix H.6. In the other direction the spectral term helps both lifts, by 1.86% for the frame lift and 0.79% for the spherical-coordinate lift, so neither effect is a proxy for the other.

Coherence and sharpness, measured separately. At each degree the error power decomposes as $E_{\text{err}} = E_{\text{pred}} + E_{\text{target}} - 2\sqrt{E_{\text{pred}} E_{\text{target}}}\,\chi$, where E_{pred}, E_{target} and E_{err} are the band-summed power of the prediction, of the target and of the error, so a ratio of error powers mixes how much of the truth a forecast captures with how sharp it is. We therefore measure coherence, $\chi = (E_{\text{pred}} + E_{\text{target}} - E_{\text{err}})/(2\sqrt{E_{\text{pred}} E_{\text{target}}})$, which isolates the first, and find that the spectral objective raises it by up to 8% at degrees above 24 and by at most 0.4% at degrees below 9, so its effect is confined to the fine scales. Sharpness is checked separately through the amplitude calibration $\sqrt{E_{\text{pred}}/E_{\text{target}}}/\chi$, which is 1 for a forecast damped to the conditional-mean optimum of (20) and above 1 for one carrying more amplitude than its coherence justifies. Both objectives are strongly over-sharp at fine scales and long lead, the accumulated rollout noise of Figure 9: at $168\,\text{h}$ and $\ell = 49$–64 the spherical-coordinate lift calibrates at 4.23 under the spectral objective against 4.52 under squared error, and at $240\,\text{h}$ and $\ell = 25$–48 at 6.44 against 6.75. The spectral objective is closer to 1 in every fine-scale long-lead cell of the spherical-coordinate lift and learned-embedding pairs, so it carries less unsupported structure, not more.

Stability. Under plain squared error every augmented learned-embedding run collapsed, and under the spectral objective only axial RoPE with $\mathrm{SO}(3)$ augmentation did. Appendix H.13 gives the two mechanisms and the changes that remove them.