

Planetary-Scale Shear Trajectories and Their Expression in Global Geological Geometry

Craig Stone

27 December 2025

Plain Language Summary

Many large geological features on Earth—such as sweeping mountain arcs, curved continental margins, rift systems, ocean-floor ridges, and sedimentary belts—share recurring geometric patterns that appear across regional, continental, and sometimes near-hemispheric scales. These features are normally interpreted in terms of local processes such as plate convergence, mantle flow, erosion, or rifting. In this study, we ask a broader question: whether some aspects of this large-scale curvature and directional organisation may also reflect an underlying, planet-scale stress framework that interacts with—but does not replace—conventional tectonic processes.

To explore this idea, we compare observed stress orientations from the World Stress Map with a mathematically defined global shear field derived from an idealised, true-polar-wander-like rotational geometry. Rather than asking only whether the model reduces global average misfit between predicted and observed stress directions, we examine whether the *pattern* of agreement and disagreement is spatially organised in a meaningful way.

Using permutation-based spatial statistics, we find that misfit values are not randomly distributed. Instead, they form coherent geographic clusters across multiple spatial scales, from regional domains hundreds of kilometres across to broad, near-hemispheric regions spanning several thousand kilometres. This organisation persists even though the model does not outperform randomly rotated alternatives in terms of global mean misfit. The result shows that spatial organisation and global alignment are distinct diagnostic signals, and that a model may capture physically meaningful structure even when it does not minimise average misfit.

We also show that the same shear geometry is expressed qualitatively in a wide range of geological and geomorphic settings, including passive-margin sediment belts, arcuate mountain systems, curved rift segments, ocean-floor bathymetric arcs, and glacially sculpted landscapes. These expressions occur across different tectonic regimes

and geological ages, suggesting that local processes may operate within, and be subtly conditioned by, a longer-wavelength stress architecture.

A key limitation of this study is that demonstrating spatial organisation is easier than demonstrating its cause. The results do not prove that true polar wander—or any specific mechanism—is responsible for the observed patterns. Instead, they support a more cautious interpretation: that a global-scale shear framework produces testable geometric signals that are detectable in present-day stress orientations and in the long-term curvature of major geological features. In this sense, the work should be viewed as a hypothesis-driven investigation rather than a causal reconstruction.

Overall, the findings suggest that regional tectonics and plate processes do not operate in isolation, but may be influenced by a persistent, planet-scale stress topology that shapes where and how large-scale curvature develops. Later sections of the paper outline how this hypothesis can be tested further using independent datasets, alternative stress models, and additional quantitative metrics.

Abstract

Large arcuate geological features occur across Earth's surface in sedimentary, tectonic, glacial, and oceanic environments. Most examples are interpreted primarily in terms of local or regional processes such as plate convergence, rifting, mantle flow, or differential erosion. However, the recurrence of smooth, large-radius curvature across contrasting geological provinces raises the question of whether some aspects of Earth's surface geometry may also reflect an underlying, long-wavelength stress organisation that interacts with—but does not replace—conventional tectonic mechanisms.

In this study, we compare global geological and geophysical patterns with an analytically derived surface shear field associated with a prescribed class of true-polar-wander-like rotational geometries. Using a Vening

Meinesz-style formulation, two conjugate families of shear trajectories (Net 1 and Net 2), together with invariant contours, are computed analytically on the sphere without tuning to observations. These trajectories are then evaluated against present-day stress orientations from the World Stress Map (WSM) and against the large-scale curvature of geological and geomorphic features.

Quantitative comparison shows that, although the modeled shear field does not outperform randomly rotated Euler-field ensembles in terms of global mean angular misfit, the spatial distribution of misfit exhibits statistically robust, scale-dependent geographic organisation. Permutation-based spatial autocorrelation analysis yields consistently positive and highly significant Moran’s I values from regional (~ 250 km) through continental to near-hemispherical (~ 3000 – 4000 km) scales, with smoothly decreasing magnitude at increasing wavelength. This demonstrates that spatial organisation and global alignment constitute distinct diagnostic signals, and that physically meaningful structure can be present even when average misfit metrics remain non-diagnostic.

Qualitative comparison further shows recurring geometric congruence between the modeled shear trajectories and a wide range of Earth-surface features, including passive-margin sediment belts, arcuate orogenic systems, rift curvature, oceanic bathymetric arcs, and glacially sculpted landscapes. These correspondences persist across differing tectonic settings and geological ages, suggesting that regional processes may operate within a persistent, long-wavelength stress framework.

Seismic anisotropy studies further indicate that upper-mantle deformation beneath continents is coherent, persistent, and often fossilized over geological timescales [Silver (1996)]. These observations provide independent support for the plausibility of a long-wavelength stress or shear framework capable of imprinting stable directional organization into the lithosphere–upper mantle system. While such anisotropy does not uniquely constrain the geometry or mechanism of the inferred shear field, its documented spatial coherence is consistent with the scale-persistent organization detected in the stress–misfit analysis.

Independent validation using the SEISGLOB2 global shear-velocity tomography model reveals a depth-localised geometric association between high-amplitude mid-mantle anomalies and the Euler domains predicted by the shear-geometry framework, confined to the 900–1200 km depth interval.

A key limitation of this work is that demonstrating spatial organisation is inherently easier than demonstrating causation. The results do not establish a unique phys-

ical mechanism for the inferred shear topology, nor do they imply that true polar wander is the sole or necessary driver. Rather, they support the interpretation that a global-scale shear framework produces testable geometric and statistical signals that warrant further investigation.

Taken together, the results indicate that Earth’s deformation and surface geometry may be influenced by a persistent, planet-scale stress topology that interacts with, and subtly conditions, plate-tectonic and regional geological processes. The framework developed here provides a basis for future tests using independent datasets, alternative stress models, and forward geodynamic simulations.

1 Introduction

Earth’s lithosphere exhibits pervasive curvature at regional, continental, and planetary scales. Arcuate sedimentary belts, sweeping mountain chains, curved passive margins, bathymetric arcs, rift systems, and coherent basin chains occur across diverse tectonic environments and geological ages. These features are conventionally interpreted in terms of local or regional boundary conditions, including plate convergence, rift propagation, mantle flow, differential erosion, and structural inheritance (e.g. [Zoback, 1992; Heidbach et al., 2018; Stephan, 2023]). While such processes explain many aspects of individual systems, some of the largest arcuate structures cross multiple geological provinces and persist across contrasting lithospheric ages, exhibiting smooth, large-radius curvature that is not always readily reconciled with short-wavelength forcing or segmented tectonic histories.

These observations motivate a broader question: whether certain aspects of Earth’s surface geometry and stress orientation may also reflect long-wavelength, planet-scale stress organisation that interacts with—but does not replace—plate-tectonic and regional processes. Previous studies have characterised large-scale coherence in observed stress orientations and have compared numerical or geodynamic stress models to the World Stress Map (WSM) primarily using scalar misfit metrics or regional smoothing approaches (e.g. [Heidbach et al., 2018; Cao et al., 2021; Stephan, 2023]). However, relatively little work has examined whether the *geographic organisation* of misfit between a prescribed global stress geometry and observed stress orientations contains additional diagnostic information beyond global mean alignment.

The present study evaluates whether a mathematically prescribed global shear field, derived from a simple class of true-polar-wander-like (TPW-like) rotational geometries, exhibits meaningful correspondence with both observed

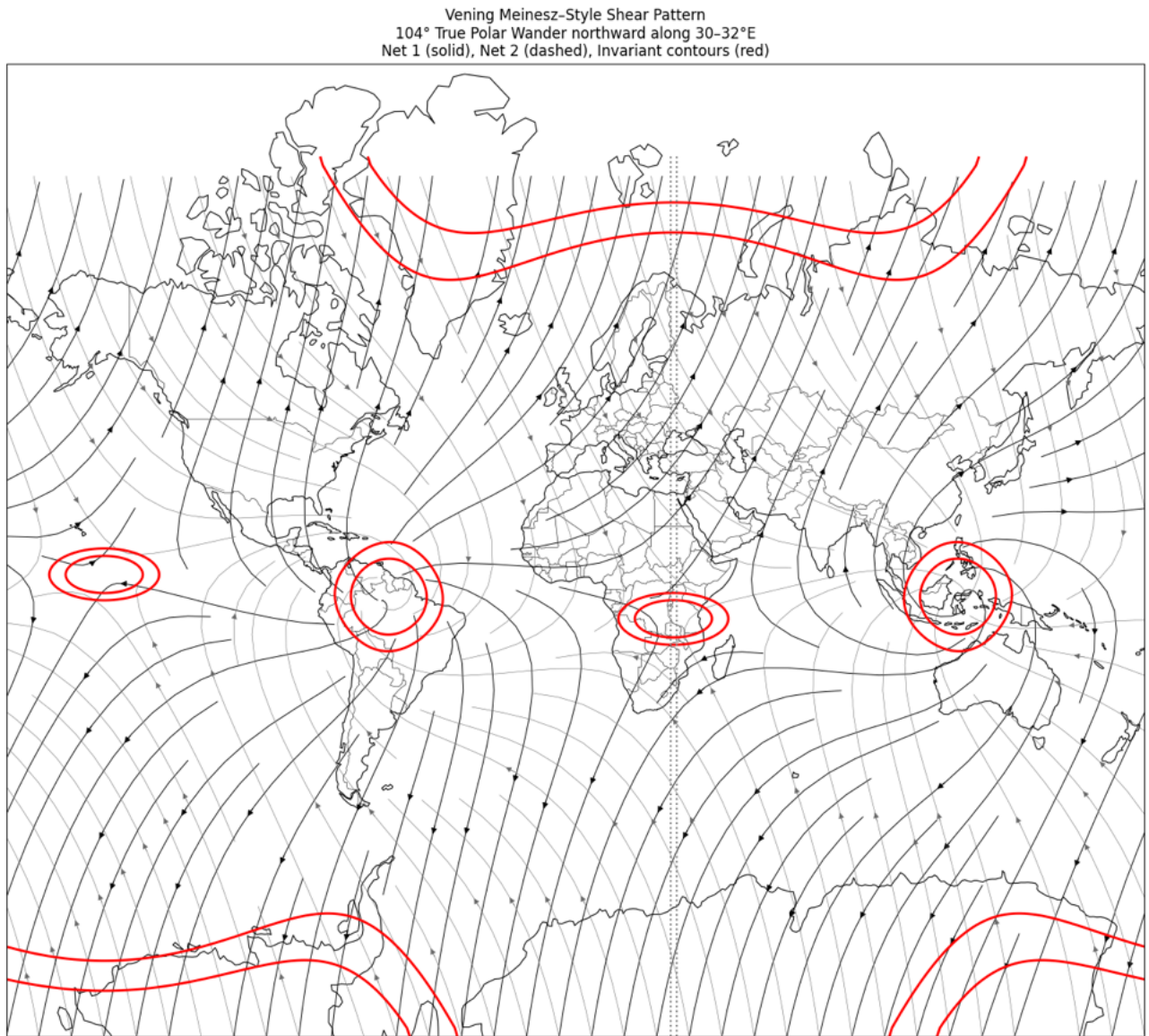


Figure 1: Vening Meinesz-style global shear pattern for a prescribed 104° northward true polar wander along the 31°E meridian. Solid curves indicate Net 1 shear trajectories, dashed curves Net 2 trajectories, and red curves invariant contours.

stress orientations and the recurring curvature of large-scale geological and geomorphic features. The approach does not attempt to reconstruct a specific geodynamic mechanism or geological event. Instead, it treats the shear topology as a *geometric hypothesis* and asks whether its predicted structure is detectable in (i) the spatial organisation of stress–orientation misfit and (ii) the geometry of large-scale surface curvature.

A central methodological distinction in this work is the separation of two diagnostic questions. First, we assess whether the modeled shear field reduces global mean angular misfit relative to randomly rotated reference fields. Second—and more critically—we evaluate whether the geographic distribution of misfit values is spatially organised rather than randomly dispersed. These quantities diagnose different aspects of model behaviour: a geometry may fail to minimise global misfit yet still express coherent long-wavelength spatial structure. Distinguishing between global alignment and spatial organisation is therefore essential for correct interpretation (cf. [Stephan, 2023](#); [Cao et al., 2021](#)).

To address these questions, we compute analytically derived shear trajectories and invariant contours on the sphere using a Vening Meinesz–style kinematic formulation and compare these trajectories with observed stress orientations from the WSM ([Heidbach et al., 2018](#)). Spatial organisation is evaluated using permutation-based Moran’s I statistics across multiple spatial scales, in parallel with Euler-rotation null ensembles that preserve internal geometry while removing Earth-fixed alignment. Expanded mathematical development of the shear-field computation and statistical testing workflow is presented in Section [2](#).

An independent motivation for considering long-wavelength stress organization comes from seismic anisotropy. Global syntheses of SKS shear-wave splitting demonstrate that upper-mantle anisotropy is widespread, coherent over hundreds to thousands of kilometres, and commonly preserved as fossilized lithospheric fabric rather than reflecting instantaneous asthenospheric flow ([Silver, 1996](#)). Fast polarization directions frequently align with major orogenic belts, ancient structural grain, and long-lived deformation corridors, indicating persistence of organized strain geometry over geological timescales. While seismic anisotropy does not uniquely define a global shear topology or its driving mechanism, its scale, coherence, and longevity establish the physical plausibility of a stress framework capable of imprinting stable directional organization—an expectation tested quantitatively in later sections through spatial statistics and axial misfit analysis.

The approach is intentionally framed to test for geometric organisation rather than to assert a unique physical mechanism, recognising that in geophysics, kinematic coherence may be demonstrable well before its ultimate dynamical explanation.

2 Methods

2.1 Methodological Context

The objective of this study is to test whether a mathematically prescribed, planet-scale shear topology produces spatially organised correspondence with independently observed stress orientations and large-scale geological curvature. Prior work has documented regional to continental organisation in the observed stress field and has evaluated numerical or geodynamic stress models against the WSM using scalar misfit metrics, smoothing radii, or regional averaging approaches (e.g. [Zoback, 1992](#); [Heidbach et al., 2018](#); [Cao et al., 2021](#); [Stephan, 2023](#)). The present approach extends this literature by (i) deriving an analytically closed-form global shear field on the sphere and (ii) explicitly evaluating the *spatial organisation* of the resulting misfit field using permutation-based Moran’s I across multiple wavelengths, interpreted jointly with Euler-rotation null ensembles.

2.2 Rotation Geometry

The shear field is derived from a rigid-body reorientation of the lithosphere through a prescribed TPW-like rotation of magnitude $\Theta = 104^\circ$ about an axis along the 31°E meridian. The transformation is purely geometric and Earth-fixed; no assumptions are made regarding the timing, forcing, or physical plausibility of the rotation pathway. The Euler-pole formulation follows standard spherical kinematics (e.g. [Turcotte and Schubert, 2014](#)), and the rotation operator is written

$$\mathbf{x}' = \mathbf{R}(\hat{\mathbf{p}}, \Theta) \mathbf{x}, \quad (1)$$

where $\hat{\mathbf{p}}$ is the unit vector normal to the rotation axis.

Throughout the following analyses, references to TPW-like geometry should be read as shorthand for a class of long-wavelength rotational shear fields rather than as an assertion of a specific Earth-history trajectory. The statistical and geometric tests employed here are insensitive to the physical origin of the shear field and evaluate only whether its analytically defined topology leaves detectable spatial signatures in independent datasets.

Interpretive scope of the rotation geometry. The prescribed true-polar-wander-like rotation used in this study is introduced strictly as a *kinematic parameterization* for generating a closed-form, planet-scale shear topology on the sphere. It is not advanced as a reconstruction of a specific geodynamic event, nor as a required or exclusive physical mechanism. The hard core of the present framework is the existence of a long-wavelength, Earth-fixed shear geometry derivable from spherical rotational kinematics; the particular choice of rotation axis and amplitude serves only to instantiate one analytically tractable member of this geometric class.

Alternative rotational parameterizations, including different Euler poles, amplitudes, or even non-TPW kinematic constructions, would generate homologous shear families with comparable topological properties. Consequently, any empirical correspondence identified here should be interpreted as evidence for sensitivity to the underlying shear topology rather than as support for a specific polar-wander history. This distinction is essential to the methodological scope of the study.

2.3 Kinematic Shear Formulation

Following a Vening Meinesz-style treatment of rotational strain on the sphere (e.g. [Vening Meinesz, 1947](#), [Stewart, 1980](#)), surface points are embedded as unit vectors \mathbf{x} on S^2 . The angular velocity vector

$$\boldsymbol{\Omega} = \dot{\Theta} \hat{\mathbf{p}} \quad (2)$$

induces the surface velocity field

$$\mathbf{v}(\mathbf{x}) = \boldsymbol{\Omega} \times \mathbf{x}. \quad (3)$$

The velocity gradient is decomposed into symmetric and antisymmetric components,

$$\nabla \mathbf{v} = \mathbf{E} + \mathbf{W}, \quad (4)$$

where \mathbf{E} defines the local strain-rate tensor. Its eigenvectors $\mathbf{e}_1, \mathbf{e}_2$ define the orthogonal conjugate shear directions (*Net 1* and *Net 2*), and the magnitude of differential shear is

$$\Delta\sigma = |\lambda_1 - \lambda_2|. \quad (5)$$

2.4 Trajectory Integration and Invariant Contours

Shear trajectories are computed as integral curves of the principal-direction fields,

$$\frac{d\mathbf{x}}{ds} = \mathbf{e}_i(\mathbf{x}), \quad i \in \{1, 2\}, \quad (6)$$

subject to $\|\mathbf{x}\| = 1$. Invariant contours are defined as loci of locally minimal differential shear (cf. [Stewart, 1980](#)). All fields are generated analytically from the prescribed geometry without tuning to geological observations.

2.5 Observed Stress Data and Misfit Computation

Observed stress orientations are taken from the 2025 World Stress Map database ([Heidbach et al., 2018](#)), using standard quality weighting and axial (180°-periodic) orientation treatment. Misfit is defined as

$$\delta = \min(|\alpha_{\text{model}} - \alpha_{\text{obs}}|, 90^\circ - |\alpha_{\text{model}} - \alpha_{\text{obs}}|), \quad (7)$$

with the better-fitting of Net 1 / Net 2 retained at each site.

2.6 Statistical Testing Framework

Two complementary statistical tests are applied.

Global-Mean Alignment (Euler-Rotation Null).

Random Euler rotations preserve internal geometry while destroying Earth-fixed alignment; the observed global mean misfit is compared against the empirical null distribution (approach consistent with [Cao et al., 2021](#)).

Spatial Organisation (Permutation Moran's I).

Angular misfit values are permuted among observation locations while preserving sampling geometry and distribution, and Moran's I is computed across 250–4000 km neighbourhoods following standard spatial-statistics practice (e.g. [Cliff and Ord, 1981](#)). This test evaluates whether misfit values exhibit significant geographic clustering independent of global-mean alignment.

Axial misfit metric and statistical null model

To quantify the geometric compatibility between observed SKS fast-axis orientations and the modeled shear-net geometry, we employ an axial angular misfit metric appropriate for bidirectional fabric orientations. Both observed and modeled azimuths are reduced modulo 180°, reflect-

ing the intrinsic axial symmetry of seismic anisotropy measurements.

For an observed SKS fast-axis orientation ϕ_i and a modeled shear azimuth θ_i evaluated at the same geographic location, the axial misfit Δ_i is defined as

$$\Delta_i = \min(|\phi_i - \theta_i|, 180^\circ - |\phi_i - \theta_i|), \quad (8)$$

such that $\Delta_i \in [0^\circ, 90^\circ]$. Under random axial alignment, the expected mean misfit is 45° .

To assess statistical significance while preserving the spatial structure of both datasets, we adopt a global-rotation null model. In this null ensemble, all observed SKS fast-axis orientations are rotated by a common random angle $\alpha \in [0^\circ, 180^\circ]$, while station locations and the modeled shear field are held fixed. This procedure maintains the geographic distribution, sampling density, and spatial correlations inherent in the observations, while destroying any systematic angular correspondence with the modeled field.

For each Monte Carlo realization, misfit angles are recomputed using the same axial metric, and summary statistics (mean misfit, median misfit, and axial variance) are evaluated. Repeating this procedure for 10^4 realizations yields null distributions against which the observed statistics are compared. Reported p -values represent the fraction of null realizations with summary statistics equal to or more extreme than the observed values.

In additional tests allowing for conjugate shear families, the modeled shear azimuth at each location is represented by two orthogonal fabric directions. In this case, the misfit for each observation is taken as the minimum axial misfit to either conjugate direction. The same minimum-selection procedure is applied identically to the null ensemble, ensuring that statistical comparisons remain consistent. This approach isolates whether observed reductions in misfit exceed those expected purely from geometric accommodation to multiple admissible fabric orientations.

This axial misfit analysis constitutes an explicit early test of Prediction 8.2, which anticipates that long-wavelength shear geometries should be detectable not only in present-day stress orientations but also in fossilized upper-mantle fabric expressed through seismic anisotropy. The use of a global-rotation null preserves spatial structure while isolating angular correspondence, ensuring that any observed reduction in misfit reflects genuine geometric compatibility rather than artefacts of sampling density or regional clustering.

Independent seismic tomography dataset The shear-wave velocity anomalies analysed in this study are taken from the global tomography model SEISGLOB2 (Durand et al., 2017). SEISGLOB2 is derived from a joint inversion of Rayleigh surface-wave phase velocities, self- and cross-coupling coefficients of spheroidal normal modes, and S, SS and ScS body-wave traveltimes, and is developed to spherical harmonic degree 40 using 21 radial spline functions. The model exhibits a well-documented reorganisation of shear-velocity structure between approximately 670 and 1500 km depth, independent of any assumptions related to the present shear-geometry framework.

2.6.1 Quantifying geometric association with Euler domains

To assess whether shear-velocity anomalies in SEISGLOB2 exhibit non-random geometric organisation relative to the Euler domains predicted by the shear-geometry framework, we employed a distance-based statistical approach rather than pointwise correlation or map overlap metrics. This choice reflects the fact that the Euler domains define extended kinematic attractor regions rather than discrete spatial features.

For each depth interval analysed, shear-velocity anomalies were identified as grid points exceeding specified percentile thresholds (90th, 95th, and 97.5th percentiles of $|\ln V_s|$) within the depth band. For every selected anomaly, the angular great-circle distance to the nearest Euler domain was computed. This yielded an empirical distribution of Euler-proximity distances for the observed data.

To evaluate the significance of this distribution, it was compared against two null models designed to destroy any preferential longitudinal or rotational organisation while preserving sampling structure. The first null randomises longitude while preserving latitude and depth, testing whether Euler proximity exceeds that expected from the latitudinal distribution of anomalies alone. The second null additionally symmetrises latitude about the equator while randomising longitude, preserving $|\text{latitude}|$ and depth but eliminating hemispheric or equatorial bias. In both cases, the null distributions retain the observed anomaly amplitudes and depth-dependent sampling density.

Geometric association was quantified by comparing the observed and null distributions using cumulative distribution functions (CDFs) of Euler-proximity distance, together with summary statistics including the mean and median distance shift relative to null expectations. A systematic leftward displacement of the observed CDF rel-

ative to both null models indicates preferential clustering of anomalies nearer to Euler domains. Robustness was assessed by examining the dependence of the effect on anomaly percentile and depth interval rather than relying on a single threshold or significance test.

2.7 Scale Definitions

Spatial scales are reported using the following conventions: regional (~ 250 – 500 km), continental to transcontinental (~ 500 – 1500 km), and planetary to near-hemispheric (~ 1500 – 4000 km). “Spatial organisation” refers specifically to statistically resolvable geographic structure in the misfit field as quantified by Moran’s I ; “global alignment” denotes scalar reductions in mean angular misfit.

3 Results: Regional Case Studies

The following case studies illustrate how the modeled shear framework is expressed across contrasting tectonic, geomorphic, and depositional environments. In each example, geometric correspondence is evaluated in terms of (i) alignment with Net 1 and Net 2 shear trajectories or invariant contours, (ii) independence from purely local boundary conditions, and (iii) coherence across scale transitions. These regional observations complement the quantitative spatial-statistical results and provide contextual evidence for a persistent, long-wavelength shear topology that interacts with, rather than replaces, plate-tectonic and surface processes.

Role of regional case studies

The regional examples presented in this section are intended as *illustrative diagnostics* rather than as independent tests of the shear framework. They are subordinate to the quantitative results reported in Sections 5–7 and are included to demonstrate how statistically detected spatial organisation manifests across contrasting geological contexts.

Accordingly, these case studies should not be interpreted as post hoc confirmations of the model. No single example is treated as decisive, and none is required for the statistical conclusions of the paper to hold. Their function is instead to show that the same analytically prescribed shear topology recurs coherently across domains governed by distinct local processes, lithologies, and tectonic histories.

3.1 Southeastern United States Sediment Arc

The arcuate sedimentary belt spanning the southeastern United States (Figure 2) represents one of the clearest expressions of the modeled global shear framework acting within a passive-margin setting. Conventional interpretations emphasise Appalachian orogenic inheritance, differential erosion, fluvial routing, and marine sediment redistribution. While these processes are important contributors, they do not readily explain the remarkable geometric continuity, large radius of curvature, and cross-provincial coherence of the arc across multiple sediment classes and lithospheric domains.

When examined in the context of the modeled shear field, the sediment arc aligns closely with invariant shear contours rather than with present-day drainage basins, coastline geometry, or individual tectonostratigraphic boundaries. The arc persists across terranes of differing lithology and age, including Paleozoic orogenic belts, Mesozoic rifted margins, and Cenozoic sedimentary platforms, indicating control by a stress geometry that transcends local structural segmentation.

Notably, the correspondence is expressed simultaneously in clay-rich sediments, carbonate distributions, and sandy deposits, each characterised by distinct transport pathways and depositional mechanisms. The convergence of these independent sedimentary signals along the same arcuate trajectory argues against coincidental alignment and instead points to a shared, long-wavelength controlling framework. Within the shear-net interpretation, the arc occupies a zone of minimal differential shear, a setting expected to favour long-term sediment accumulation and curvature stability while suppressing disruptive deformation.

Through repeated episodes of erosion, sea-level fluctuation, and sediment reworking, this inherited stress architecture appears to have been progressively amplified rather than erased. The southeastern sediment arc therefore functions as a geomorphic and sedimentological recorder of the broader shear topology, consistent with the cross-scale organisational behaviour demonstrated statistically in Section 7.

This interpretation does not displace established Appalachian or passive-margin models; instead, it situates them within a hierarchical framework in which local processes operate within, and are subtly conditioned by, a persistent planet-scale stress organization. In this sense, the southeastern United States sediment arc is not an isolated regional anomaly, but a continental-scale mani-

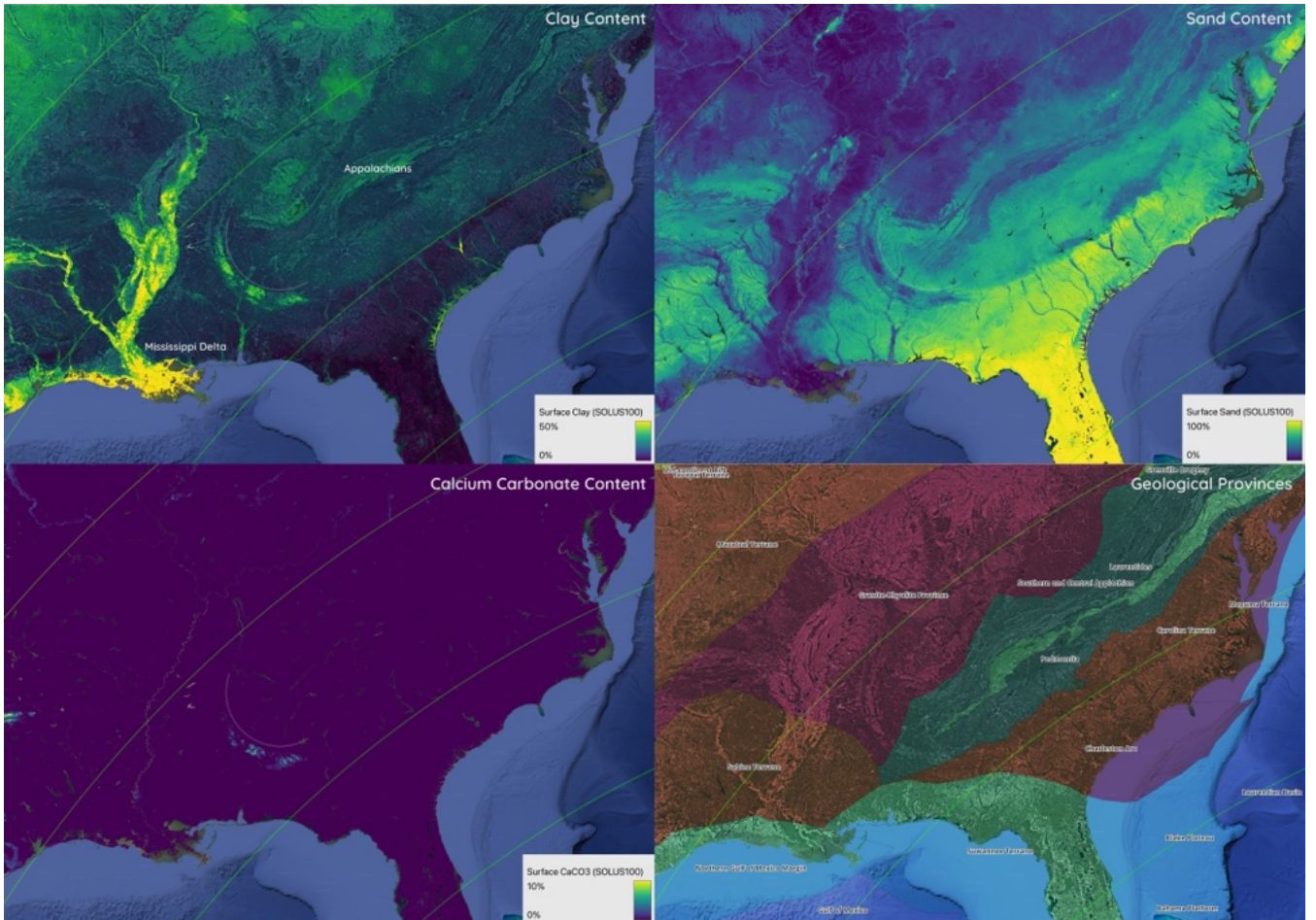


Figure 2: Four-panel comparison of clay, CaCO_3 , sand, and geological provinces in the southeastern United States. A continent-scale arcuate belt aligns with invariant shear contours and crosses multiple geological provinces.

festation of the same shear geometry expressed in other tectonic and depositional systems worldwide.

3.2 The Himalayan Arc

The Himalayan mountain belt is widely interpreted as the product of continent–continent collision between India and Eurasia, leading to crustal shortening, uplift, and thrust-belt development. This tectonic framework explains the kinematics and deformation style of the orogen, but it does not, by itself, account for the remarkable geometric regularity and large-radius curvature of the Himalayan arc over more than 2500 km.

Within the modeled global shear framework, the Himalayan arc lies within a transition zone where the two conjugate shear families rotate smoothly along strike. The arc exhibits near-orthogonality to one family and approximate parallelism to the other, producing a mechanically coherent curvature geometry. In such a configuration, collision-driven shortening is predisposed to organise into laterally continuous curvature rather than into segmented salients or irregular thrust corridors, consistent with observations of arc-scale continuity in major Himalayan structures.

This interpretation does not replace plate-tectonic explanations for Himalayan evolution. Instead, it frames the arc as a case in which strong collisional forcing operates within a long-wavelength geometric environment that biases curvature development. The behaviour is consistent with the broader pattern observed across other arcuate systems in this study, in which large-radius curvature aligns with regions of predicted shear stability while remaining dependent on local tectonic drivers.

3.3 The Caribbean Plate Margin and Intra-Arc Curvature

The Caribbean region is shaped by a combination of subduction, strike-slip motion, slab fragmentation, and arc–continent interaction along the boundaries of the Caribbean Plate (e.g. ??). Arc geometry and forearc curvature vary along strike, reflecting regional changes in plate coupling, trench kinematics, and crustal inheritance (Rosencrantz et al., 1990).

Within the modeled shear framework, several major Caribbean arc segments occupy domains where one shear family remains locally stable over 1000–1500 km while the conjugate family rotates across the plate margin. Curved forearc and intra-arc basins develop preferentially in regions where shear trajectories converge or change orientation gradually rather than abruptly. In contrast, highly

oblique or transform-dominated sectors coincide with domains of rapidly rotating shear geometry and correspondingly larger stress–orientation misfit.

This behaviour is interpreted as consistent with the idea that plate-boundary deformation is primarily controlled by subduction and strike-slip processes, while the long-wavelength shear topology influences the geometric form through which curvature and segmentation are expressed along the margin.

3.4 South Indian Ocean Tectonic Arc

The southern Indian Ocean hosts one of the least discussed yet most geometrically coherent arcuate tectonic features identified in this study. Extending across abyssal plains and intersecting multiple spreading systems, the arc is commonly attributed to combinations of ridge propagation, transform segmentation, and inherited mantle structure. While these processes contribute to its development, they do not, on their own, account for the smooth, large-radius curvature and persistence of the feature across regions with markedly different spreading histories.

Within the modeled global shear framework, the South Indian Ocean arc aligns closely with invariant shear contours, occupying a domain of minimal differential shear under the prescribed rotational geometry. This alignment is particularly significant because oceanic lithosphere is continuously generated. The preservation of coherent curvature in such a setting implies that the organizing stress geometry is both long-wavelength and repeatedly imposed through successive episodes of lithospheric accretion.

Bathymetric and gravity data indicate that ridge segments, fracture zones, and subtle topographic highs preferentially conform to the modeled curvature rather than to local spreading directions alone. In several regions, ridge axes rotate smoothly to remain approximately orthogonal to nearby shear trajectories, while transform offsets tend to align parallel to one of the two shear families. This behaviour mirrors that observed along the Mid-Atlantic Ridge and supports the interpretation that ridge segmentation and propagation are biased by an externally imposed shear topology rather than emerging solely from local kinematic constraints.

The diagnostic value of this feature lies in its predominantly oceanic context, largely free from continental inheritance. Its morphology must therefore arise from interactions among mantle flow, lithospheric accretion, and a persistent, planet-scale stress organization. In this setting, the global shear field provides a parsimonious explanation for the spatial coherence of ridge initiation, abandonment,

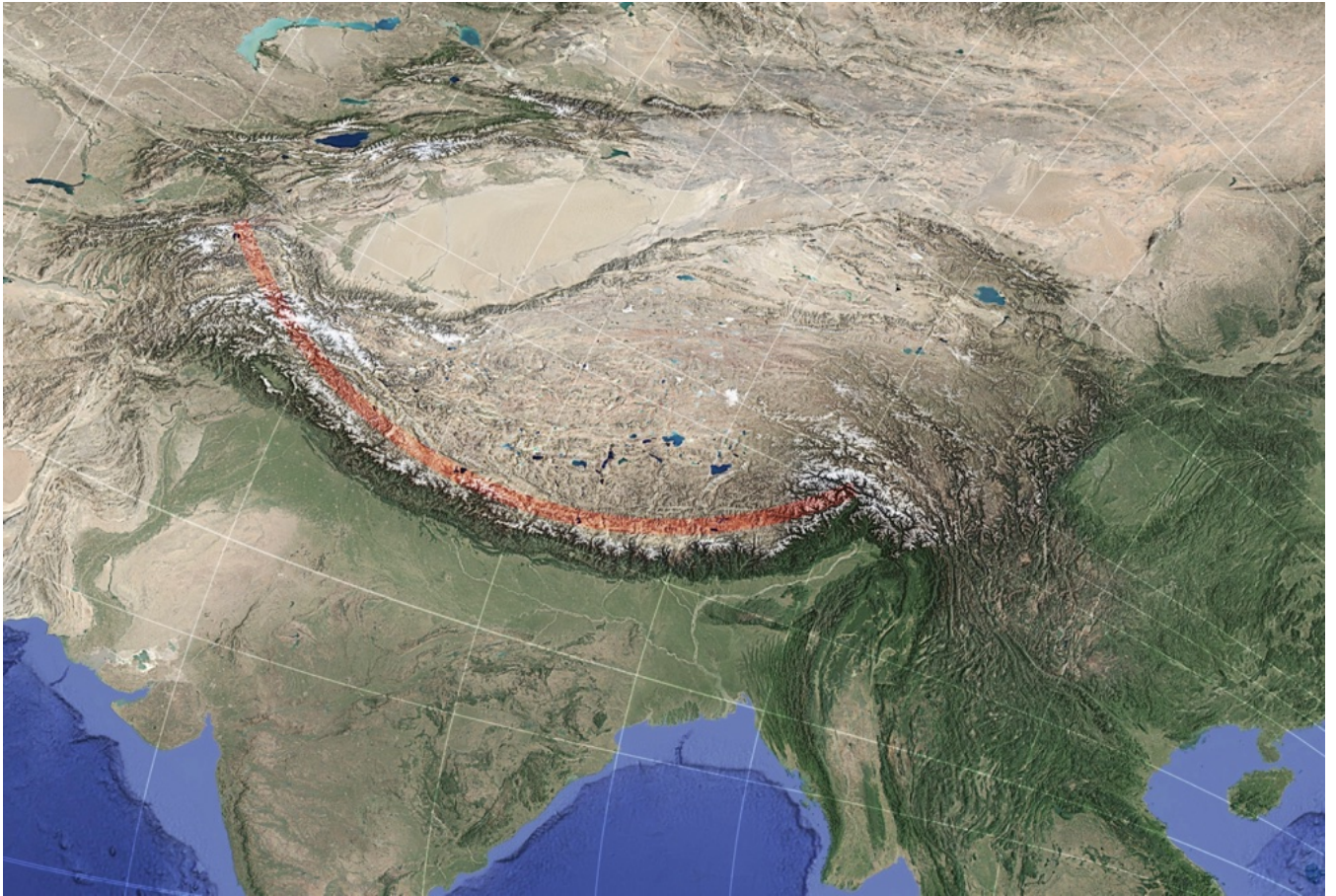


Figure 3: The Himalayan mountain belt forms a geometrically smooth arc that lies parallel to one shear family and orthogonal to the conjugate family.

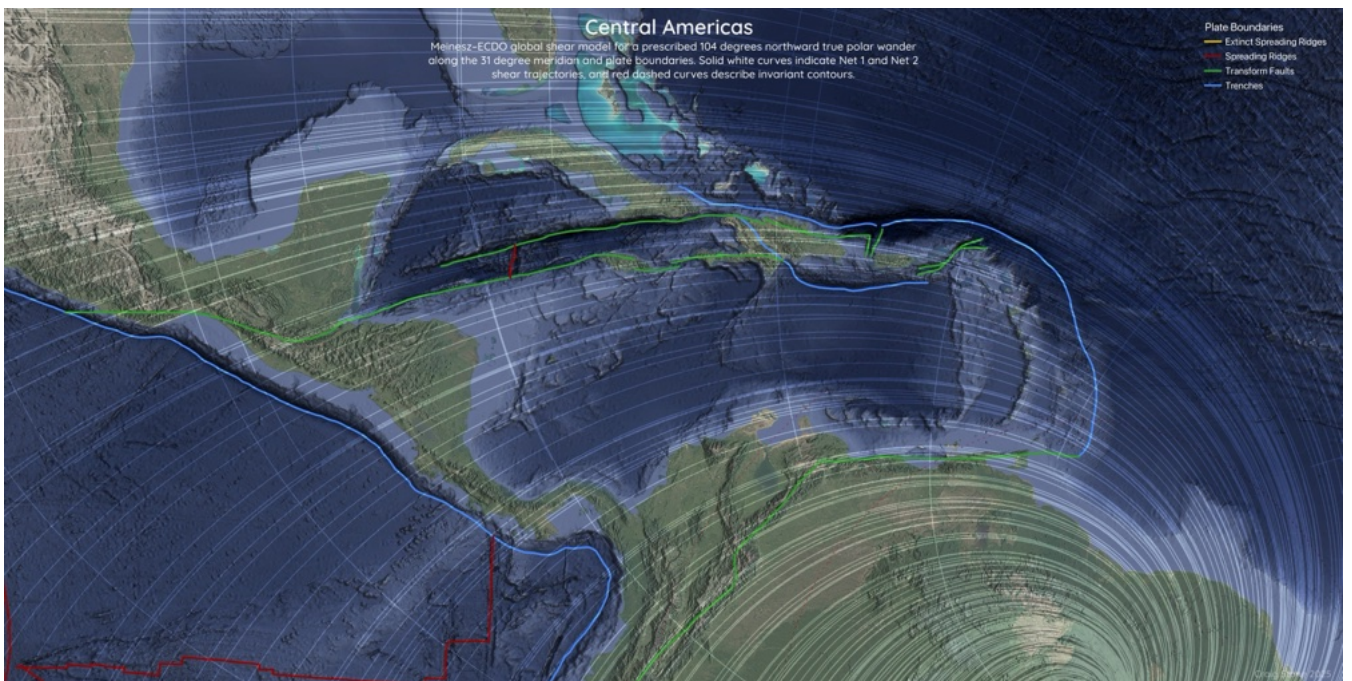


Figure 4: Central America and the Caribbean showing strong geometric congruence between plate boundaries and the modeled global shear field.

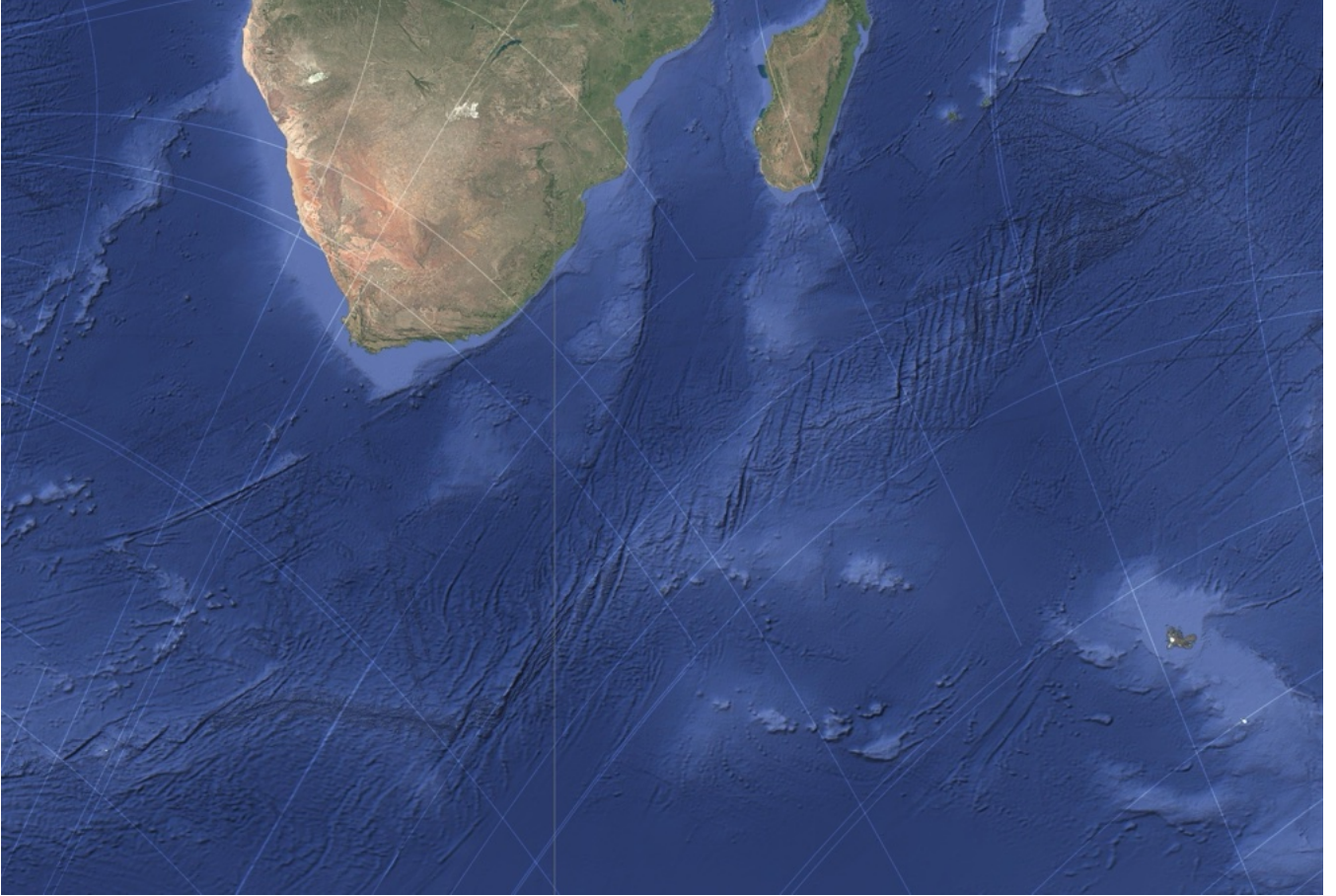


Figure 5: A vast tectonic arc in the southern Indian Ocean displaying strong congruence with modeled shear trajectories.

and segmentation through time.

Even as spreading centres migrate and plate configurations evolve, the invariant shear geometry remains fixed relative to the global frame, enabling successive generations of oceanic crust to inherit similar curvature. The resulting composite bathymetric signature integrates the cumulative effect of long-wavelength shear organisation over tens of millions of years—behaviour that is consistent with the scale-persistent spatial structure demonstrated statistically in Section 7.

In summary, the South Indian Ocean tectonic arc represents a compelling oceanic manifestation of the global shear framework. Its correspondence with invariant contours strengthens the inference that the modeled shear field reflects a real and long-lived component of Earth’s stress architecture, influencing both continental and oceanic tectonics across multiple spatial and temporal scales.

3.5 The Banda–Sulawesi Arc System

The Banda–Sulawesi region is characterised by complex arc–continent collision, microplate rotation, slab roll-back, and highly curved orogenic belts (e.g. [Hamilton, 1979](#);

[Hall, 2012](#)). Numerous studies attribute arcuate geometry in this region to progressive rotation of lithospheric blocks and trench-rollback processes in a convergent–collisional setting ([Hall, 2012](#)).

In the context of the modeled shear field, the Banda Arc occupies a domain in which the two shear families rotate through nearly orthogonal orientations over relatively short distances. The arc follows a curvature envelope that remains close to one of the shear trajectories, while lateral structural terminations approach invariant-contour zones associated with low differential shear. This configuration is consistent with the persistence of sweeping arc curvature through multiple stages of collision and rotation.

The present interpretation does not replace rotation-based or rollback-driven models of Banda evolution. Instead, it suggests that the persistence and smoothness of the arc geometry may be favoured where long-wavelength shear trajectories provide a kinematic pathway that supports sustained curvature development during otherwise complex plate interactions.

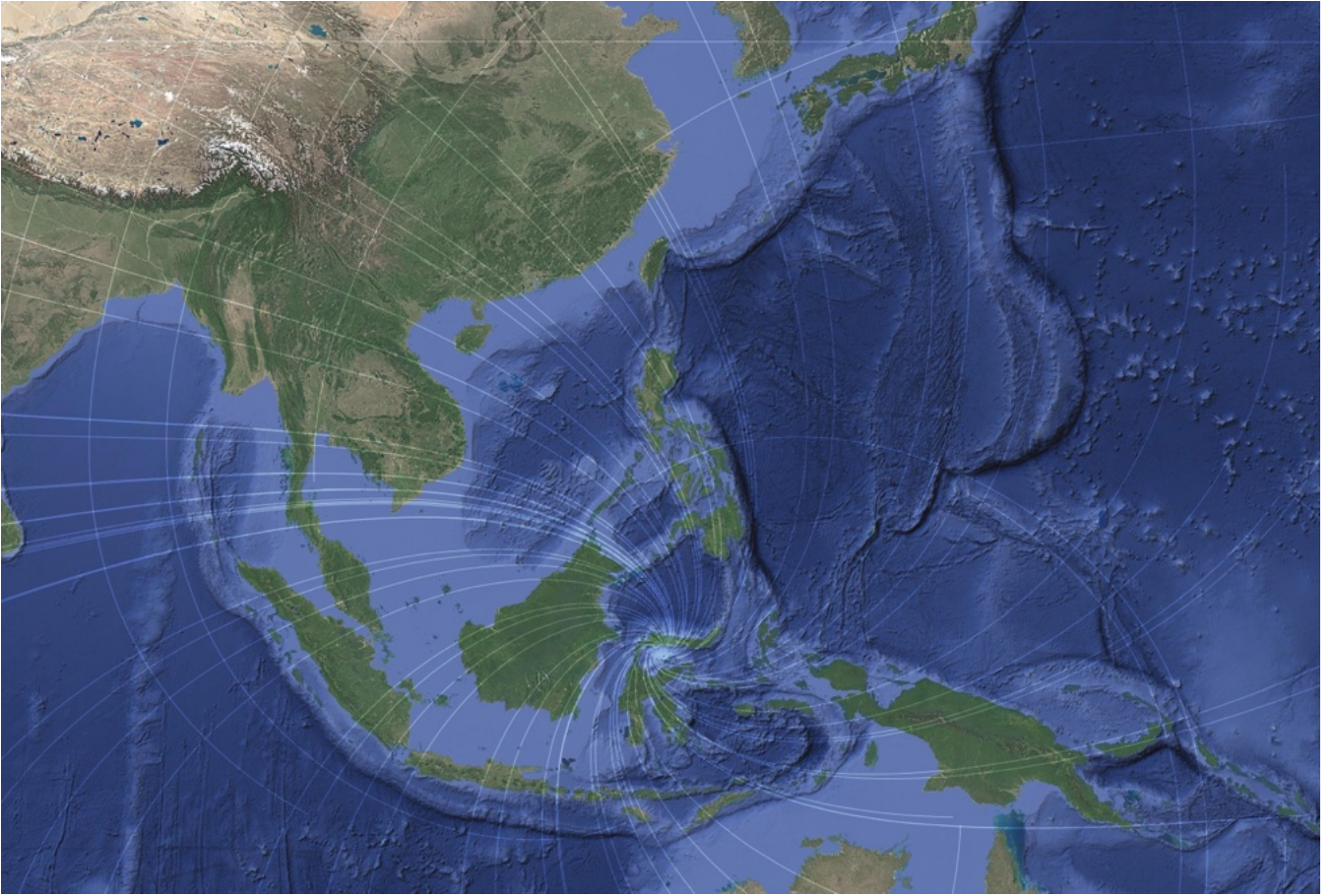


Figure 6: Primary and secondary bathymetric arcs near Sulawesi aligned with predicted shear trajectories.

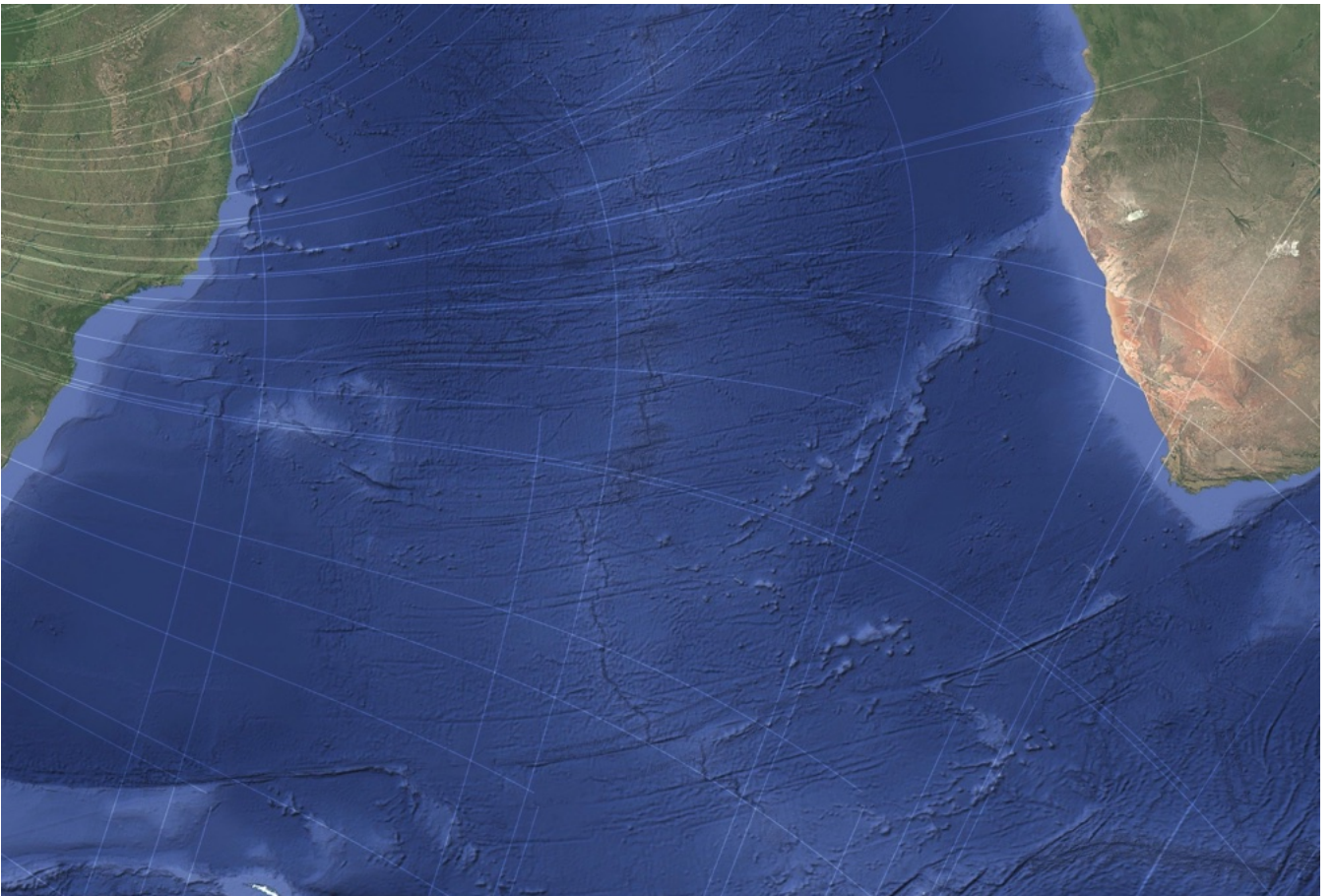


Figure 7: The Mid-Atlantic Ridge exhibits alternating alignment with Net 1 and Net 2 shear families along strike.

3.6 Mid-Atlantic Ridge

The morphology of the Mid-Atlantic Ridge (MAR) reflects the interaction of plate divergence, mantle upwelling, and transform segmentation (e.g. [Macdonald, 1984](#); [Parsons and Sclater, 1977](#)). Along-strike alternation in ridge-segment orientation and curvature has been attributed to variations in spreading kinematics and thermal structure (?).

When viewed in the context of the modeled shear framework, MAR segments exhibit recurring alternation between orientations approximately orthogonal to one shear family and parallel to the conjugate family. This pattern is expressed at multiple wavelengths and persists across both hemispheres. The correspondence is interpreted not as evidence that the shear field controls ridge dynamics, but as an indication that spreading-related segmentation may preferentially exploit orientations that are geometrically consistent with a long-wavelength stress topology.

This behaviour parallels that observed in several continental rift systems, where rift curvature and segmentation occur preferentially within predicted shear-trajectory domains. In both cases, local geodynamic processes remain primary, while the modeled shear geometry appears to bias the form through which segmentation is expressed.

3.7 North American Intracontinental Basins and Lake Systems

Several large intracontinental basins and lake systems of North America, including the Hudson Bay depression, the Great Lakes region, and adjacent basin chains, have histories linked to lithospheric flexure, glacial loading and unloading, sedimentary subsidence, and inherited structural fabrics (e.g. ??). These processes explain basin formation mechanisms, but the large-radius curvature and directional coherence of basin chains extend beyond individual structural blocks.

When evaluated against the modeled shear framework, basin margins and lake basins display preferential alignment with one or both of the shear families, with directional alternation expressed at multiple wavelengths. Regions of curvature stability coincide with predicted low-shear or invariant-contour domains, while zones of strong curvature change tend to occur where the two shear families rotate rapidly or intersect.

In this interpretation, glacial excavation, flexural rebound, and sedimentary loading provide the dominant mechanical processes, while the long-wavelength shear topology biases which orientations persist and accumulate over geological time. The behaviour is consistent with the

broader cross-continental pattern in which geomorphic and tectonic curvature appears preferentially where the modeled framework predicts coherent shear geometry.

3.8 British–Irish Geomorphic and Structural Bimodality

Across Britain and Ireland, large-scale landscape and structural patterns exhibit a pronounced bimodality in orientation and curvature, expressed in drainage networks, valley alignments, structural lineaments, and morphotectonic grain (e.g. [Ballantyne, 2002](#); [Woodcock and Strachan, 2014](#)). Many of these features reflect the interaction of glacial modification, inherited basement structure, and later tectonic reactivation ([Phillips et al., 2016](#)).

When evaluated in the context of the modeled shear field, the dominant directional sets correspond closely to the two conjugate shear families, with alternation between them expressed at regional to trans-regional scales. Curvature stability and valley persistence are concentrated within predicted low-shear domains, while abrupt directional switching occurs near regions of rapid shear-trajectory rotation.

In this setting, glacial erosion, fluvial incision, and structural reactivation provide the principal geomorphic and tectonic processes. The long-wavelength shear framework is interpreted as a geometric environment that biases which orientations are preferentially preserved, re-excavated, or reactivated through successive climatic and tectonic cycles, consistent with observations of repeated lineament reuse in the British–Irish region (e.g. [Woodcock and Strachan, 2014](#); [Johnson, 1997](#)).

3.9 East African Rift System

The East African Rift System (EARS) represents an active zone of continental extension associated with lithospheric thinning, magmatism, and rift-segment development from the Afar triple junction to Mozambique (e.g. [Ebinger, 1989](#); [Taylor et al., 2013](#); [Corti, 2009](#)). Rift geometry varies along strike, with alternating straight and arcuate segments and systematic changes in rift-basin orientation through time ([Corti, 2009](#)).

Within the modeled shear framework, the principal rift corridors lie within domains where one shear family remains approximately parallel to the rift axis while the conjugate family rotates across the rift margin. Arcuate rift curvature and segment bifurcation occur preferentially in regions where the two shear families converge or rotate rapidly along strike. In these zones, local extensional

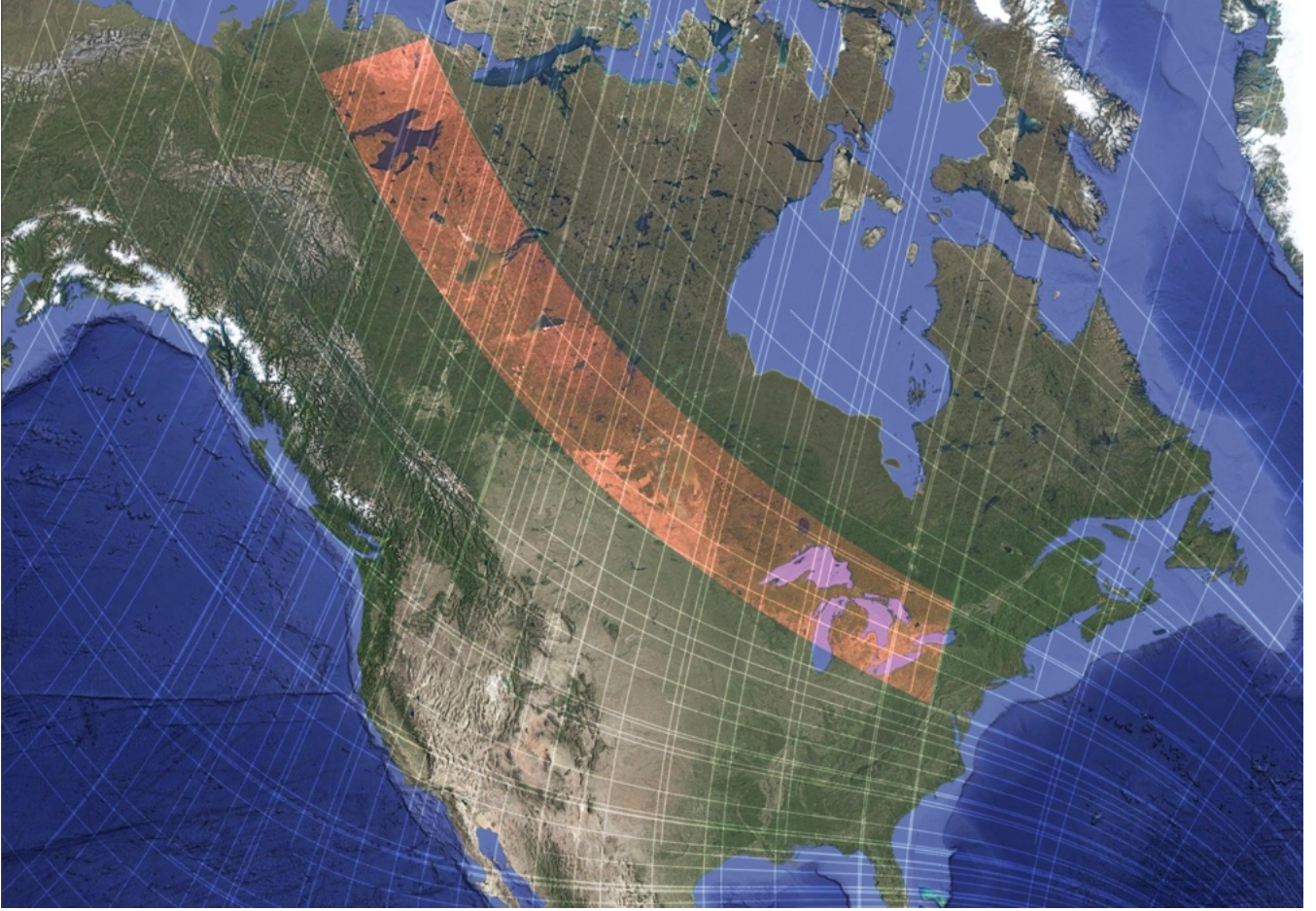


Figure 8: A continental-scale arc of large lakes across North America aligned with invariant shear contours.

processes may be predisposed to organise into sweeping curvature rather than linear fault segments.

This interpretation is consistent with observations that rift propagation and transfer-zone development are strongly influenced by pre-existing anisotropy and stress-field orientation (e.g. Corti, 2009; Philippon and Corti, 2015).

3.10 The Greenland–Hudson Bay Region

The Greenland–Hudson Bay sector records a long history of Precambrian craton assembly, lithospheric stabilization, glacial loading and unloading, and intracontinental basin development (e.g. Philippon and Corti, 2015; Corti, 2009). The Hudson Bay depression and surrounding structural corridors display large-radius curvature and concentric basin geometry that have been linked primarily to glacial isostatic adjustment and crustal flexure (?).

When compared with the modeled shear framework, major basin margins, reactivated structural corridors, and regional lineament sets occur preferentially along one or both shear-trajectory families, with alternating alignment expressed at regional to continental scales. Domains of curvature stability coincide with predicted low-shear regions,

whereas directional switching and structural dispersion occur near zones of rapid shear-trajectory rotation.

In this interpretation, glacial and flexural processes remain the dominant mechanisms responsible for basin formation and landscape modification. The long-wavelength shear framework is treated as an organising geometry that helps explain the persistence and directional coherence of curvature across structurally heterogeneous crustal provinces, consistent with evidence for repeated structural reactivation in the Canadian Shield and adjacent regions (e.g. Percival et al., 2012).

3.11 Northwestern Canada and the Fort Simpson Arc

The Fort Simpson Arc anomaly (FSA) of northwestern Canada has traditionally been interpreted as an Early Proterozoic magmatic arc, based on its curvilinear aeromagnetic signature, associated positive gravity anomalies, and U–Pb zircon ages clustering near ~ 1.85 Ga from granitoid basement exposures. This interpretation, while well supported by geochronological and petrological evidence, encounters several persistent geometric and tectonic ambiguities that warrant re-examination in light of the global



Figure 9: Bimodal geomorphic orientation in the United Kingdom, particularly Scotland, aligned with both shear families.

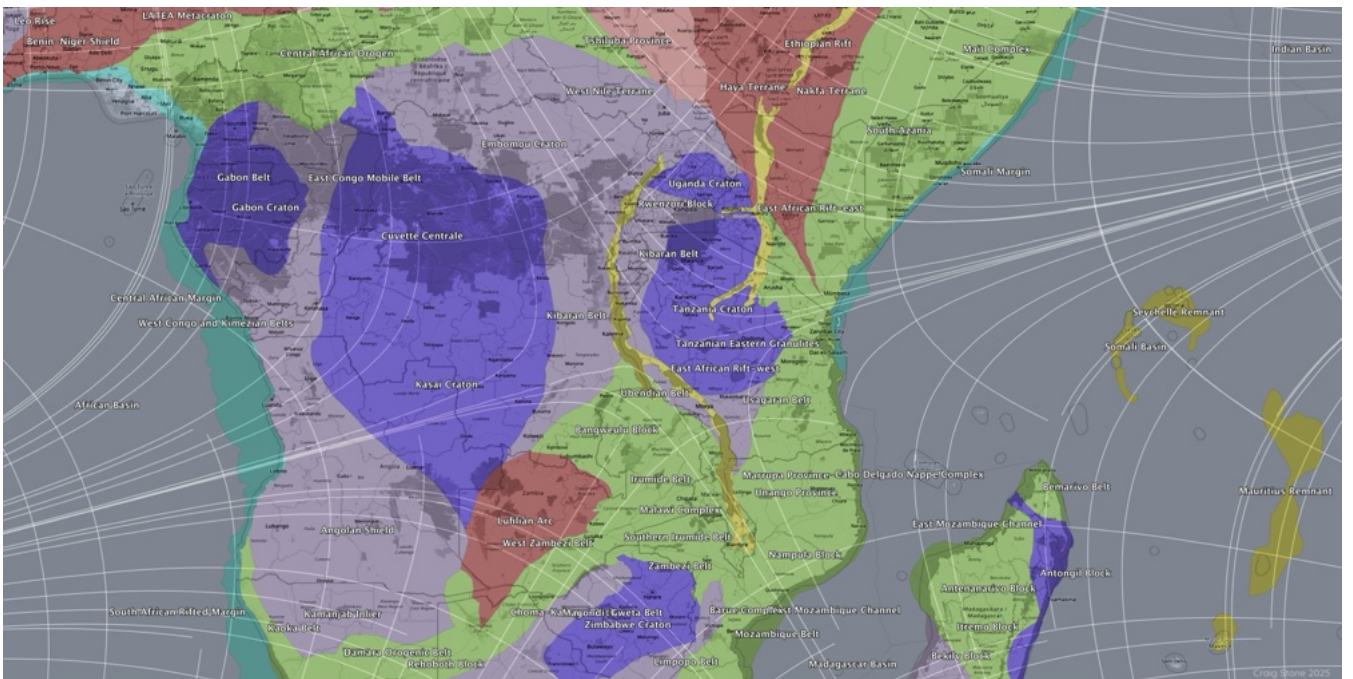


Figure 10: Double-arc geometry of the western East African Rift coincident with convergence of Net 1 and Net 2 near 14°S, 31°E.

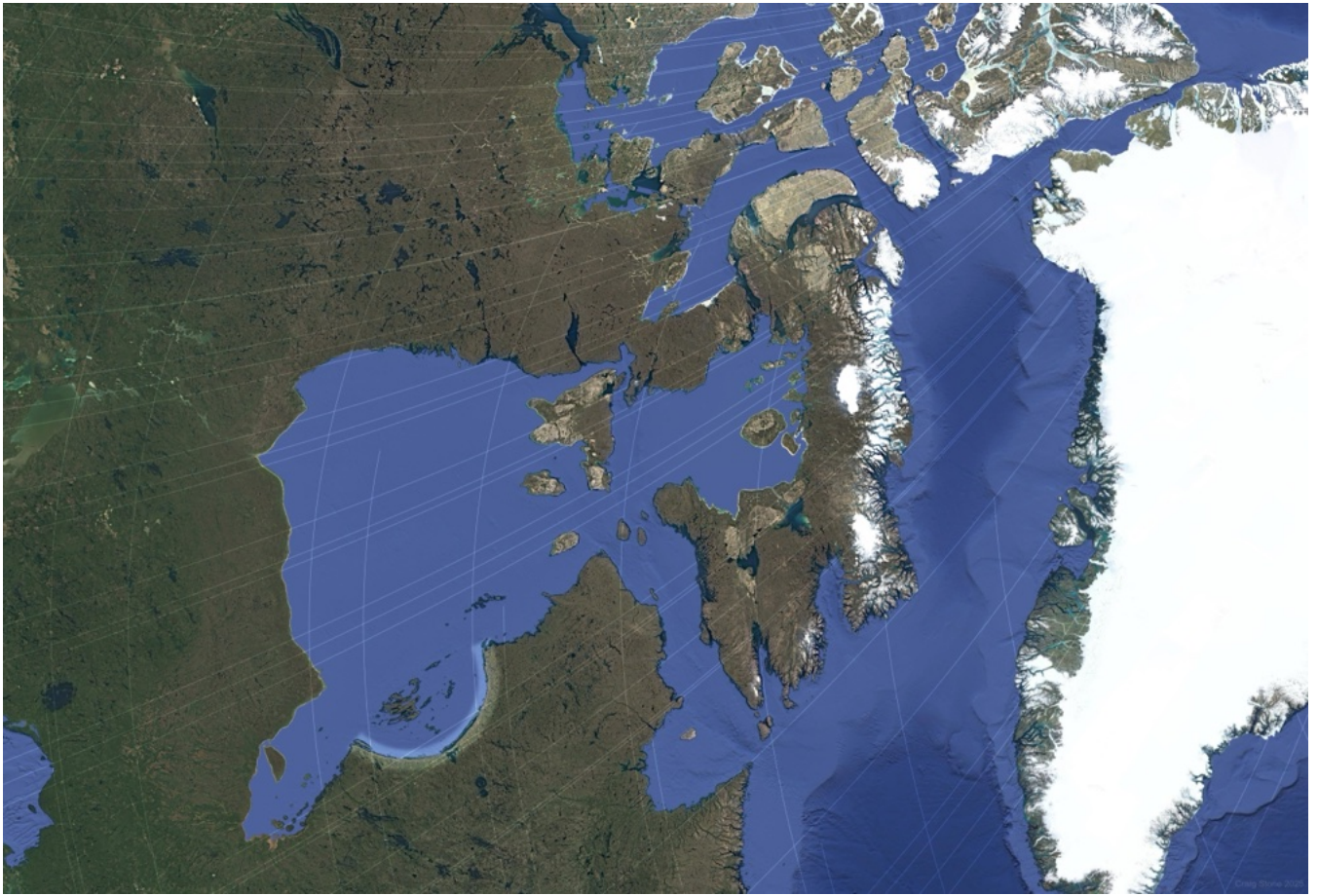


Figure 11: Greenland and Hudson Bay in polar stereographic projection showing bimodal agreement with shear nets and the arcuate Nastapoka structure.

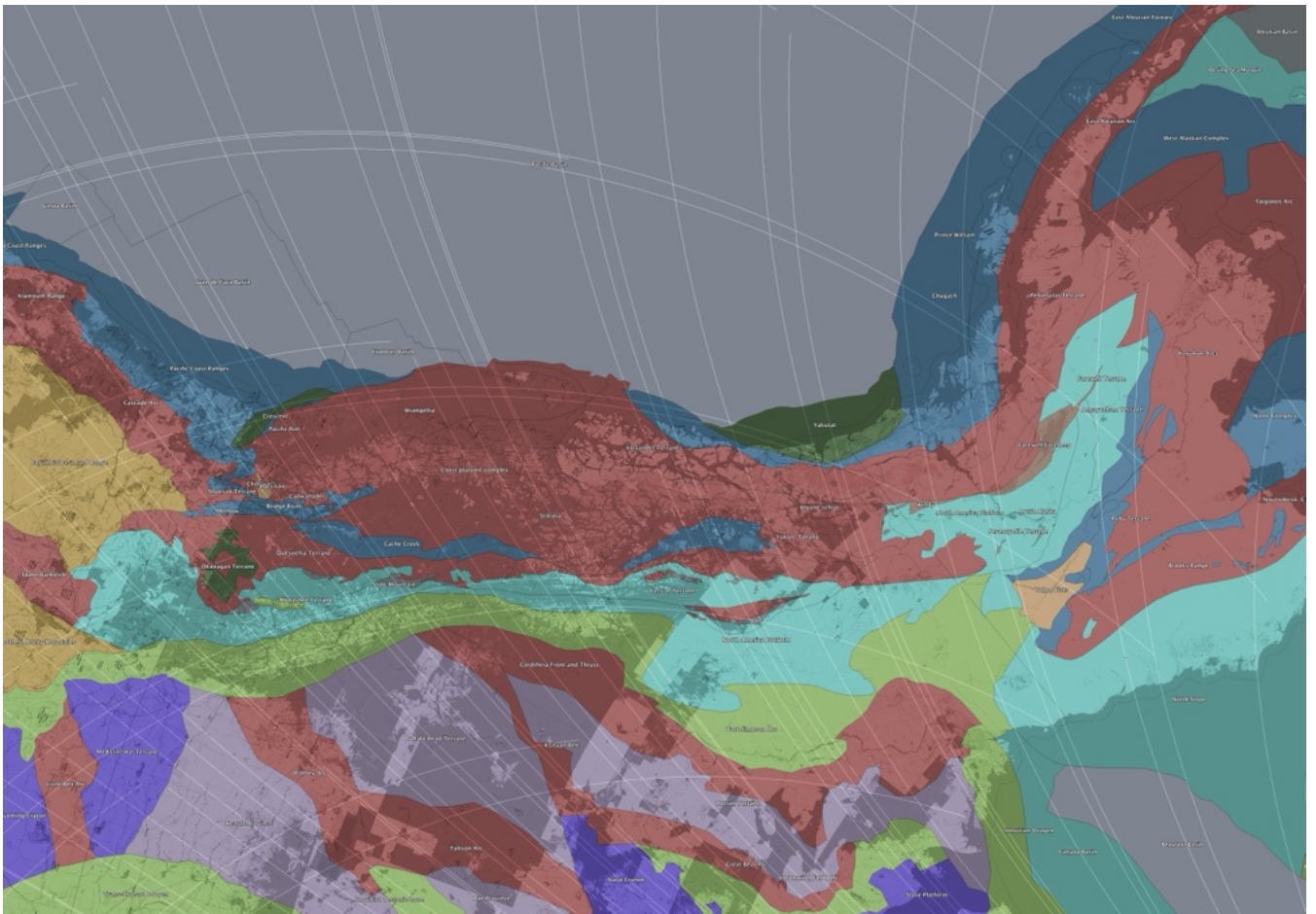


Figure 12: Northwestern Canada showing multiple agreements with shear nets, including the Fort Simpson Arc and transition from the Coast Plutonic Complex to the East Aleutian Arc.

shear-net framework developed here.

Previous syntheses note that the Fort Simpson anomaly is not a simple linear feature, but instead broadens and curves westward into large elliptical magnetic highs near Great Bear Lake, among the most extensive aeromagnetic features in North America. Even within conventional interpretations, these western highs are acknowledged as potentially distinct basement elements that appear unified only through filtered geophysical datasets. Moreover, the Fort Simpson domain exhibits a long history of reactivation, influencing later extensional basins, dyke swarms, hydrothermal systems, and lithospheric segmentation over a time span exceeding one billion years.

When examined in polar and Arctic projections, the Fort Simpson Arc displays a close geometric correspondence with one of the principal branches of the modeled global shear net. In particular, the curvature, bifurcation, and apparent bimodality of the Fort Simpson structure coincide with zones of net convergence and rotation in the shear trajectories. This correspondence is independent of stratigraphic boundaries, terrane affinities, or magmatic age constraints, suggesting a kinematic control that predates and outlasts any single tectono-magmatic episode.

Within this framework, Early Proterozoic arc magmatism is reinterpreted not as the origin of the Fort Simpson structure, but as one expression of a pre-existing lithospheric shear corridor. Such corridors are expected to localize mantle upwelling, magma ascent, and crustal underplating during periods of favorable stress orientation, while remaining mechanically weak and prone to reactivation under subsequent stress regimes. The coincidence of the Fort Simpson anomaly with positive gravity signatures is consistent with shear-focused lower-crustal densification or mafic underplating, rather than requiring a uniquely arc-related mechanism.

This reinterpretation resolves several longstanding difficulties associated with an arc-only model. The pronounced curvature and elliptical geometry of the Fort Simpson anomaly, its apparent continuity despite internal geological heterogeneity, and its repeated reactivation through Proterozoic and Phanerozoic time are all natural consequences of long-lived lithospheric shear structures, but are atypical of transient subduction-related arcs. Importantly, this perspective does not negate the arc-related geochemical and geochronological evidence documented in earlier studies; rather, it reframes arc magmatism as a secondary response to an inherited stress architecture.

In this view, the Fort Simpson Arc anomaly represents a lithospheric-scale shear feature that has acted as a persistent organizer of deformation and magmatism since

at least the Paleoproterozoic. Its alignment with the global shear net supports the broader hypothesis that certain first-order continental geological features are governed by planet-scale kinematic patterns associated with large-amplitude true polar wander, rather than by isolated plate-boundary processes alone.

3.12 The North Atlantic–Arctic Sector

The North Atlantic–Arctic region records a complex history of rifting, continental breakup, seafloor spreading, and passive-margin evolution associated with opening of the North Atlantic and Arctic basins (e.g. [Doré et al., 2002](#); [Eagles and Scott, 2004](#); [Buck, 2015](#)). Large-radius curvature is present in rifted margins, shelf escarpments, and offshore basin chains extending from the Labrador Sea through Greenland and into the Eurasian Arctic.

Comparison with the modeled shear field shows that many of these arcuate margin and basin geometries align with one or both shear-trajectory families, with directional alternation expressed over scales of several thousand kilometres. Regions of reduced curvature variability and long-wavelength morphological continuity occur preferentially within predicted low-shear or invariant-contour domains, whereas irregular segmentation tends to coincide with areas where the two shear families rotate rapidly along strike.

These correspondences are not taken to imply that the shear field generated margin architecture. Rifting, magmatism, and plate rupture remain the primary drivers of North Atlantic–Arctic evolution. In the present interpretation, the modeled framework instead helps explain why curvature and segmentation recur with similar spatial organisation across multiple passive-margin systems of differing age and tectonic history.

3.13 Mid- to High-Latitude Glacial Morphology and Shear-Guided Landscape Expression

Across numerous mid- and high-latitude regions, glacially sculpted terrains exhibit recurring directional coherence that aligns with the modeled global shear framework. These expressions occur in settings ranging from formerly glaciated continental interiors to high-relief fjord systems and cratonic margins, where Pleistocene and earlier glacial activity has repeatedly interacted with inherited structural anisotropy.

In many such regions, valley networks, fjord orientations, drumlin fields, and basin alignments display systematic alternation between the two modeled shear families,

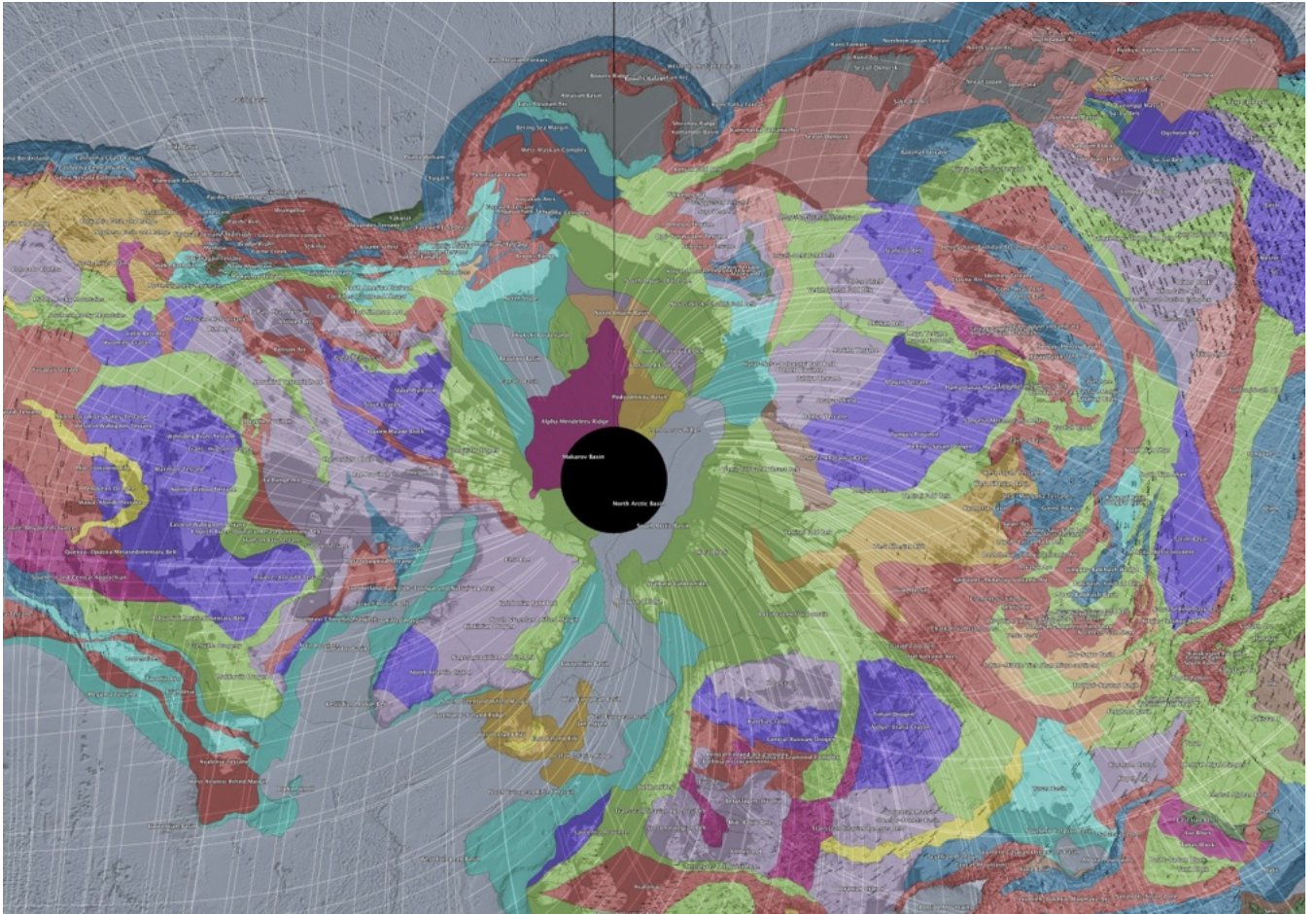


Figure 13: Arctic polar projection showing analytically derived shear trajectories overlaid on a global tectonic and geological province compilation. The shear lines exhibit strong geometric correspondence with the curvature, orientation, and boundaries of major circum-Arctic geological provinces across North America, Greenland, and Eurasia.

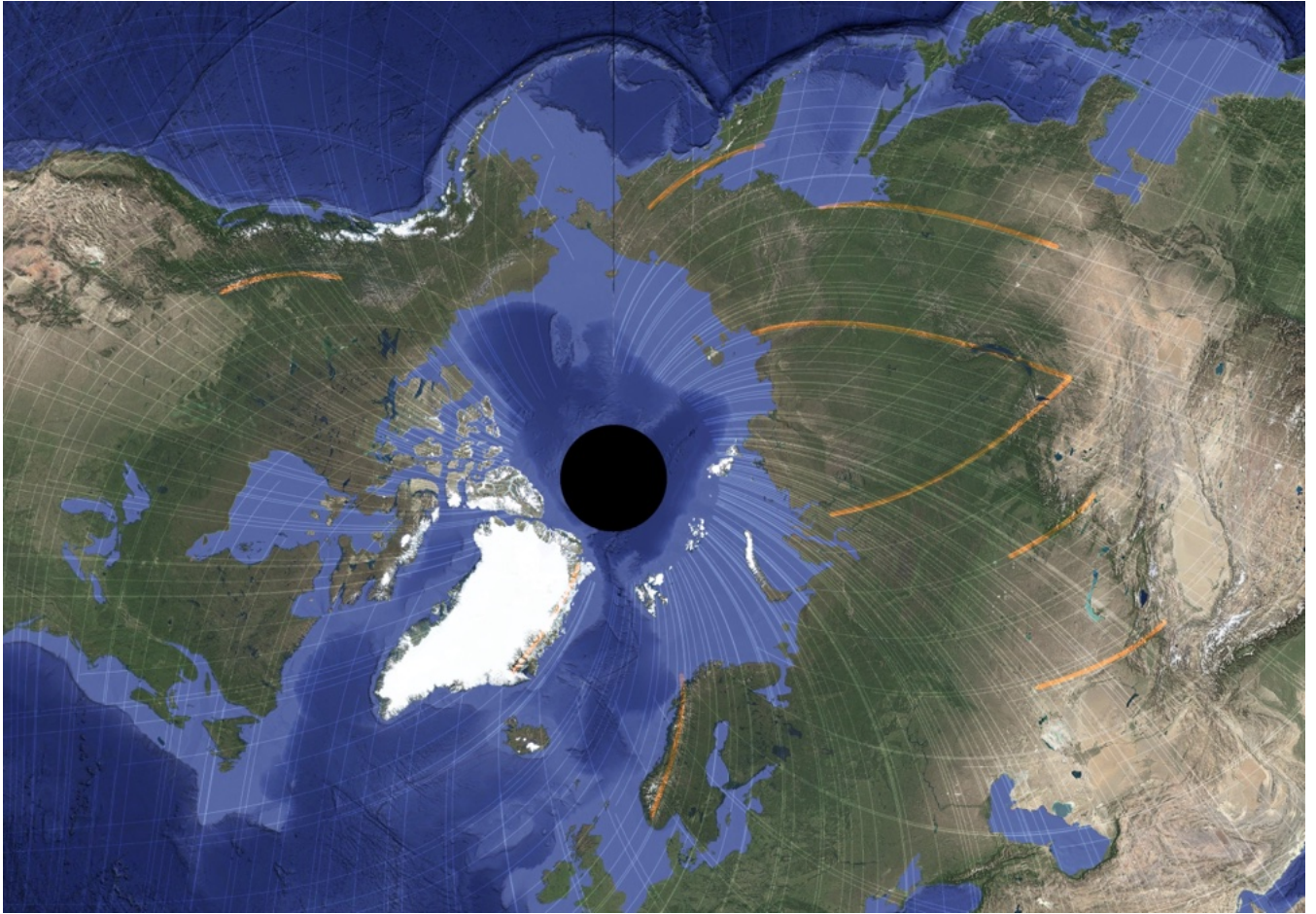


Figure 14: Arctic polar projection of shear trajectories over satellite imagery and bathymetry. Large-scale coastal arcs, continental shelf edges, submarine ridges, and basin margins frequently align with the modeled shear field, highlighting coherence across both continental and oceanic domains.

or preferential occupation of invariant-contour domains. This behaviour is observed in Scandinavia, the Canadian Shield, Greenland, coastal Patagonia, and segments of the Scottish Highlands, where glacial erosion has preferentially exploited shear-aligned weakness zones and curvature-stable domains.

These relationships are not readily explained by ice-flow dynamics alone. While ice-sheet geometry and basal processes strongly influence erosion patterns, they do not by themselves account for the persistence of large-radius curvature, repeated directional bimodality, or coherence across lithologically and tectonically contrasting terranes. Instead, the observed morphology is more consistent with selective enhancement of pre-existing, shear-guided structural fabrics through repeated cycles of glacial occupation and unloading.

Invariant-contour regions in particular tend to coincide with broad, low-strain basins and gently curving depressions that act as long-term sediment traps or loci of glacial lake development. Conversely, shear-trajectory corridors correspond to linear to curvilinear zones of enhanced incision, fjord development, and valley deepening, where stress-aligned structures provide mechanically favourable pathways for focused erosion.

This behaviour parallels that documented in the North American arcuate lake chain and in parts of Greenland and Scandinavia, where landscape anisotropy and curvature stability persist across multiple glacial cycles. The coherence of these relationships across continents and hemispheres supports interpretation of glacial morphology as an amplifying and revealing agent acting upon a longer-lived, planet-scale shear organisation rather than as an independent generator of curvature.

Taken together, these glacially expressed examples reinforce a broader pattern evident throughout the case studies: regions of persistent curvature, directional bimodality, and organised anisotropy tend to occur where the modeled field predicts stable stress geometry across scales. This behaviour is consistent with the statistically significant spatial organisation of misfit demonstrated in Section 7, further linking geomorphic expression to the same long-wavelength stress topology inferred from the stress-orientation analyses.

3.14 Large-Radius Ocean-Basin Curvature Provinces

Beyond plate-boundary systems, large-radius curvature is also expressed in bathymetric arcs, abyssal fabric trends, and basin-scale morphological sweeps across several ocean

basins, including sectors of the South Atlantic, Indian Ocean, and southwest Pacific. These features arise from a combination of spreading-ridge segmentation, transform-fault organisation, sedimentary infill, and thermal subsidence (e.g. [Parsons and Sclater, 1977](#); [Macdonald, 1984](#)).

Across these regions, the orientation of basin-scale curvature frequently alternates between the two modeled shear families, and long-wavelength continuity is most pronounced within predicted low-shear domains. Where shear-trajectory rotation is rapid, curvature becomes segmented or irregular, paralleling the behaviour observed in continental and passive-margin examples.

As in the other case studies, the interpretation advanced here is geometric rather than causal: spreading and sedimentary processes remain primary, while the modeled shear topology provides a potential explanation for the recurrence and spatial organisation of curvature across ocean-basin scales.

3.15 Continental Margin Curvature, Passive-Margin Arcs, and Long-Wavelength Structural Guidance

Curved passive margins and continental-shelf arcs occur along several major continental outlines, including segments of the West African margin, northern South America, the western Australian shelf, and portions of the Antarctic and Arctic continental rims. These features are commonly attributed to variations in rift propagation, diachronous breakup, sediment loading, or flexural response to differential margin subsidence. While these processes play important roles in shaping margin architecture, they do not fully account for the persistence of smooth, large-radius curvature that transcends basin boundaries, stratigraphic transitions, and breakup histories.

When examined within the modeled global shear framework, many of these passive-margin arcs coincide with invariant-contour domains or trace trajectories that are parallel to one of the two shear families over distances of hundreds to thousands of kilometres. The congruence is particularly evident where margin curvature remains coherent across multiple rift episodes or across transitions from volcanic to non-volcanic margin segments, implying control by an organising stress geometry rather than by locally restricted breakup kinematics alone.

Several major continental shelves display gently sweeping curvature in shelf-edge trajectories and sediment-wedge outlines that follow shear-aligned pathways despite contrasts in sediment supply, basin subsidence history, or margin thermal state. In these cases, depositional archi-

texture appears to have evolved within a preconditioned stress topology that favoured curvature stability and long-term geometric persistence, with stratigraphic processes amplifying rather than generating the underlying planform geometry.

Where transform-bounded margin segments intersect shear-trajectory corridors, margin curvature commonly tightens and structural segmentation becomes more pronounced. Conversely, margins coincident with invariant-contour regions exhibit broad, laterally continuous curvature with reduced structural partitioning. This systematic variability is consistent with deformation and margin evolution being conditioned by spatial gradients in the long-wavelength shear field.

These relationships parallel behaviours documented in passive-margin sediment belts, arcuate rifts, and ocean-basin curvature systems elsewhere in this study. In each case, large-radius curvature and directional stability are most clearly expressed where the modeled field predicts low-variance or convergent shear geometry across scales. The recurrence of this pattern suggests that passive-margin curvature is not solely the by-product of breakup mechanics, but reflects interaction between plate separation, sedimentation, and a persistent, planet-scale stress organisation.

In this interpretation, passive margins act as long-term recorders of curvature-stable domains within the global shear topology. Their planform geometry evolves through the cumulative superposition of rifting, thermal subsidence, and sediment loading on an inherited geometric framework—behaviour consistent with the scale-persistent spatial organisation demonstrated quantitatively in Section 7.

4 Regional Expressions and Cross-Scale Geometric Consistency

The regional examples presented in this section illustrate how the modeled shear framework is expressed across contrasting tectonic, geomorphic, and depositional environments. These examples are not offered as mechanistic reconstructions; local tectonic processes such as convergence, rifting, magmatism, glaciation, and sediment routing remain essential to the formation and evolution of each system (e.g. [Zoback, 1992](#); [Heidbach et al., 2018](#)). Rather, the case studies assess whether large-radius curvature, directional bimodality, and structural persistence occur preferentially in regions where the modeled field predicts stable or convergent shear geometry.

Qualitative comparison is therefore interpreted as complementary to the spatial-statistical analysis. Whereas the statistical tests evaluate whether the misfit field exhibits non-random geographic organisation, the case studies examine whether similar organisation is expressed in large-scale geological curvature. Consistency across these independent perspectives is taken as *convergent evidence*, not proof of causation.

5 Global Plate Boundary Comparison

5.1 Comparative Behaviour of the Two Euler-Point Domains

The two Euler-point regions defined by the prescribed 104° northward true polar wander (TPW) along the 31°E meridian provide a comparative geometric framework for evaluating how the modeled global shear field is expressed across contrasting tectonic environments (Figures [16](#) and [17](#)). Although both regions exhibit clear correspondence with the Venning–Meinesz–style shear trajectories, the mode of conformance differs in ways that are geologically informative.

Western Euler Domain. The western Euler point, centered over northern South America and the Caribbean (Figure [16](#)), exhibits the strongest and most internally consistent agreement between the shear model and observed geology. In this domain, shear trajectories form tightly wrapped spiral patterns that closely parallel Precambrian cratonic margins, mobile belts, and long-lived deformation zones. Major rift and transform boundaries, earthquake epicenter distributions ($M \geq 4.5$ over the past 25 years), and active to historically active volcanic centers preferentially align tangentially to the modeled shear flow.

Notably, both ancient tectonic fabrics and present-day deformation are organized predominantly along shear-parallel trajectories rather than orthogonal to them. This indicates that the shear field has acted as a persistent organizing stress over multiple tectonic cycles, repeatedly reactivating inherited lithospheric weaknesses in geometrically consistent orientations. The high degree of spatial coherence suggests that thick, mechanically heterogeneous continental lithosphere is capable of preserving and expressing low-order global stress modes over geological timescales.

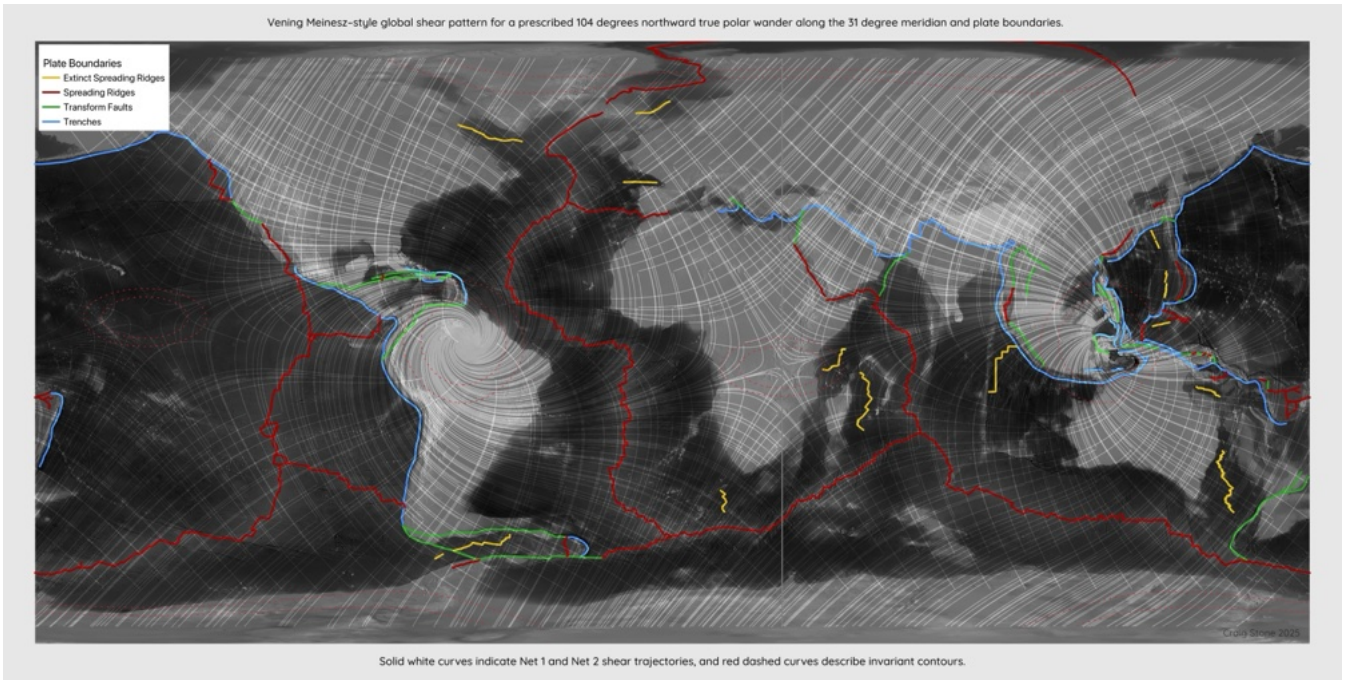


Figure 15: Comparison of the global shear model with modern plate boundaries. Strong geometric correspondence is observed across subduction arcs, transforms, and extinct ridges.

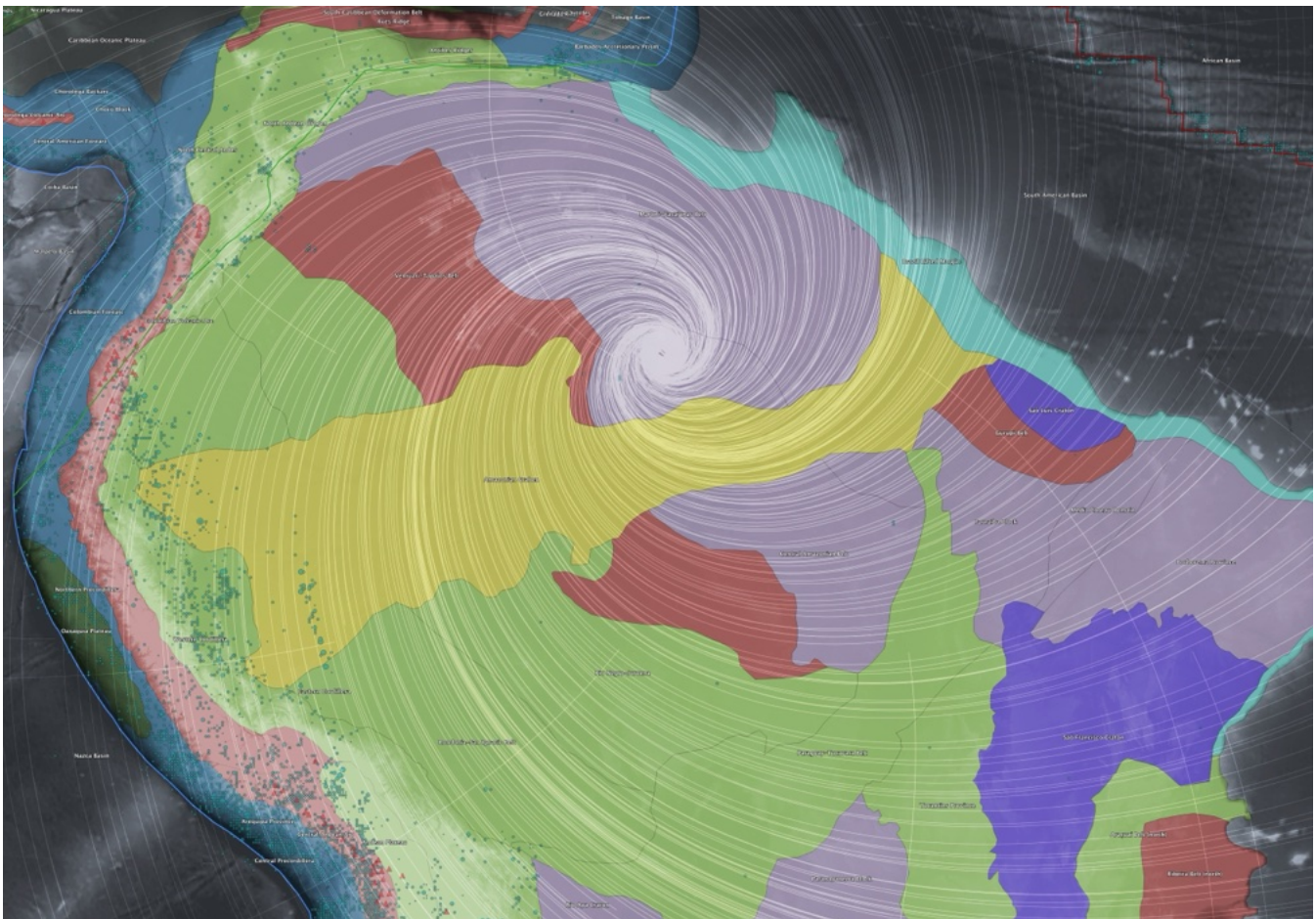


Figure 16: Western Euler-point domain showing the Vening–Meinesz-style shear trajectories overlaid on geological provinces, plate boundaries, earthquake epicenters ($M \geq 4.5$), and volcanic centers.

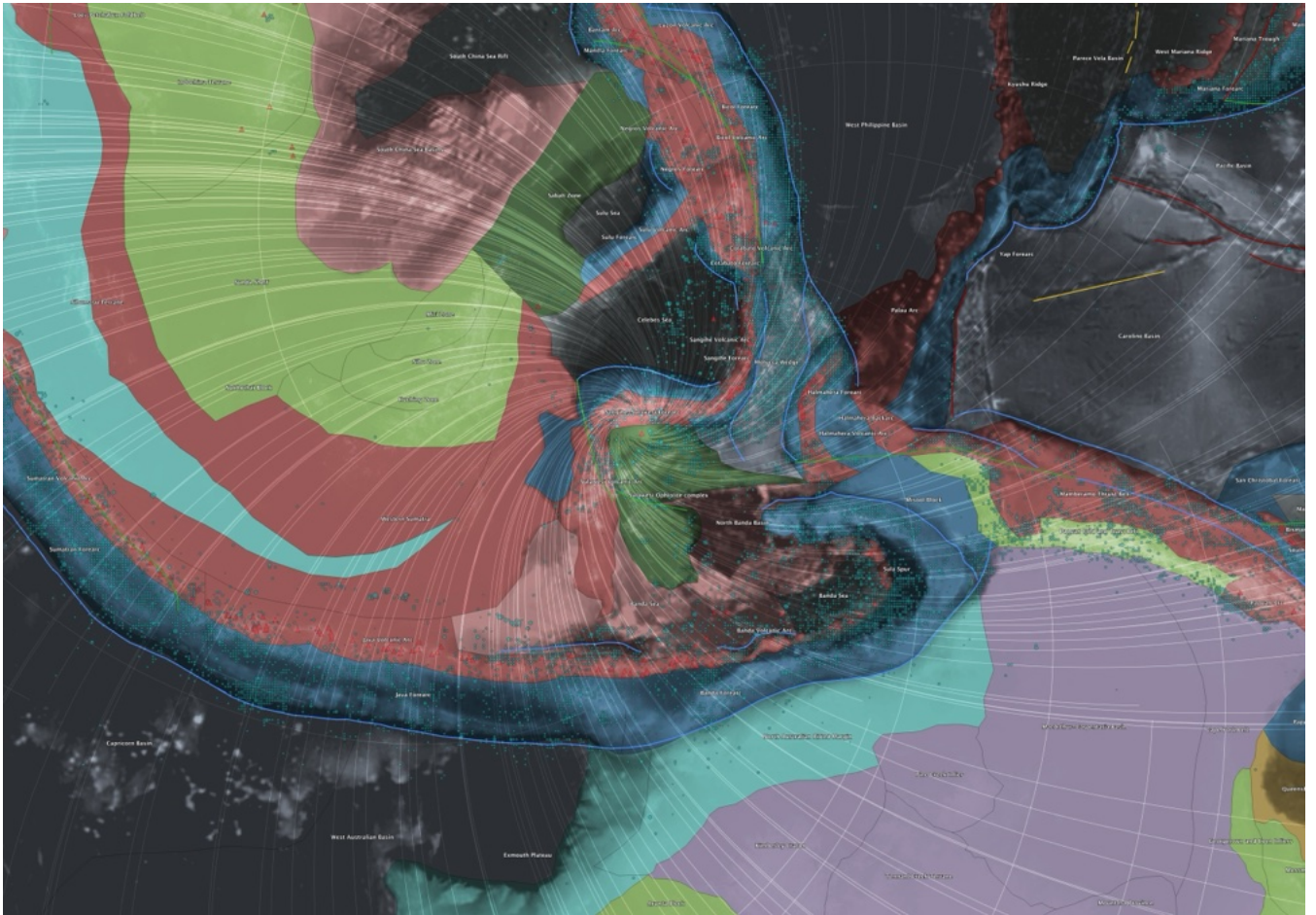


Figure 17: Eastern Euler-point domain (Indonesia–Sunda–Banda region) illustrating the relationship between modeled shear trajectories, arcuate subduction systems, seismicity, and active volcanism.

Eastern Euler Domain. The eastern Euler point, located within the Indonesia–Sunda–Banda tectonic collage (Figure 17), presents a contrasting but equally informative expression of the same underlying shear structure. This region is characterized by rapid plate convergence, slab rollback, arc–continent collision, and distributed back-arc deformation. Despite this complexity, several robust correspondences with the shear model remain evident.

Major arcuate subduction systems, including the Banda and Sunda arcs, follow curvature consistent with modeled shear trajectories rather than simple relative plate-motion vectors. Earthquake hypocenters cluster along curved bands that mirror regions of maximum shear curvature, while active and historical volcanism preferentially occupies shear-aligned arc segments and convergence zones. In this domain, rift boundaries and basin margins exhibit mixed behavior: some conform closely to shear trajectories, whereas others are dominated by local slab geometry and rollback dynamics. This indicates that while the shear field does not dictate tectonic behavior outright, it acts as a geometric constraint within which subduction-driven processes organize themselves.

Comparative Implications. The contrasting behavior of the two Euler domains highlights several important implications. First, the persistence of recognizable shear conformance in both stable continental interiors and highly active subduction zones argues against coincidental alignment and supports the presence of a global, low-order stress architecture capable of coupling into diverse lithospheric contexts. Second, regions underlain by thick, ancient lithosphere appear to preserve shear-imposed geometry more faithfully than rapidly recycling convergent margins, emphasizing the role of lithospheric memory in amplifying the visibility of the shear field.

Finally, the results indicate that the modeled shear structure should be understood not as a replacement for plate tectonics or mantle convection, but as a superimposed, long-wavelength stress mode that biases the orientation, curvature, and persistence of tectonic features generated by those processes. The systematic differences observed between the western and eastern Euler-point domains therefore strengthen the interpretation that the Venning–Meinesz–style global shear field represents a persistent and physically meaningful component of Earth’s geodynamic system.

6 Scale-Dependent Spatial Autocorrelation of the Misfit Field

Across all examined spatial scales from 250 to 4000 km, the stress–misfit field exhibits statistically significant positive spatial autocorrelation. Moran’s I decreases smoothly with increasing scale, from strong regional clustering to weaker but resolvable near-hemispheric organization. No abrupt loss of significance is observed.

This behavior indicates a finite but large correlation length for stress–misfit organization, extending well beyond individual plate boundaries or continental domains. The attenuation of Moran’s I toward hemispheric scales marks a transition from regional clustering to low-degree global structure rather than a disappearance of spatial coherence.

6.1 Short-Wavelength Structure (250 km)

At a spatial scale of 250 km, the misfit field exhibits very strong positive spatial autocorrelation, with Moran’s $I = 0.30$. The permutation null distribution has a mean effectively indistinguishable from zero and extremely small variance, yielding $p \ll 10^{-6}$. This result confirms that misfit values are highly clustered at short wavelengths and establishes a robust baseline of spatial coherence. At this scale, clustering is expected to reflect dominant local and regional tectonic controls, including fault systems, plate boundary geometry, and lithospheric heterogeneity.

6.2 Regional-Scale Coherence (500 km)

At 500 km, Moran’s I remains large ($I = 0.23$) and highly significant under permutation testing ($p \ll 10^{-6}$). The reduction in Moran’s I relative to 250 km reflects the expected decay of spatial correlation with distance, while the persistence of strong autocorrelation demonstrates that stress–misfit organization extends well beyond strictly local interactions. This scale corresponds to regional tectonic domains and large plate-boundary systems, indicating coherent stress structure at continental scales.

6.3 Long-Wavelength Organization (1000 km)

At a spatial scale of 1000 km, the misfit field continues to exhibit strong and statistically significant spatial autocorrelation, with Moran’s $I = 0.17$ and $p \ll 10^{-6}$. The persistence of coherent structure at this wavelength exceeds typical crustal or single-orogen length scales and points to long-wavelength organization of the stress field.

At this scale, purely local tectonic explanations become increasingly implausible, suggesting the influence of broader lithospheric or mantle-scale processes.

6.4 Planetary-Scale Persistence (2000 km)

Approaching the largest scales examined, 2000 km, Moran’s I remains positive and highly significant ($I = 0.11$, $p \ll 10^{-6}$). Although reduced in magnitude relative to shorter scales, the continued presence of statistically robust spatial autocorrelation at near-planetary wavelengths demonstrates that the stress–misfit field is organized over distances comparable to hemispheric dimensions. This result rules out explanations confined to regional tectonics alone and establishes a minimum correlation length of at least several thousand kilometers.

6.5 Upper-Bound Scale and Finite Correlation Length

Extending the spatial autocorrelation analysis to larger neighborhoods further clarifies the scale dependence of the stress–misfit field. At 3000 km, Moran’s I remains positive and statistically robust ($I = 0.073$, $p \ll 10^{-6}$), indicating coherent organization at near-hemispheric wavelengths. At 4000 km, the magnitude of Moran’s I is reduced but remains resolvably positive ($I = 0.050$, $p \ll 10^{-6}$), demonstrating that spatial structure persists even as neighborhood extents approach hemispheric scales.

Taken together, results from 250 to 4000 km reveal a smooth, monotonic decay of Moran’s I with increasing spatial scale, without abrupt loss of significance. This behavior indicates a finite correlation length for the stress–misfit field, rather than unbounded or scale-invariant organization. The attenuation of spatial autocorrelation toward hemispheric extents marks the transition from regional clustering to low-degree global structure and represents the practical upper limit of interpretability for distance-based spatial autocorrelation metrics such as Moran’s I .

6.6 Scale Dependence and Correlation Length

Taken together, the results reveal a smooth, monotonic decay of Moran’s I with increasing spatial scale, without loss of statistical significance across the full range examined (250–3000 km). Such behavior is characteristic of a genuine spatial correlation process with a long correlation length, rather than noise or sampling artifacts. The absence of a sharp cutoff indicates that the organizing

mechanism operates across multiple spatial regimes, from local tectonic structure to global-scale stress organization.

6.7 Global Alignment versus Spatial Structure

Two distinct statistical questions are addressed in this study: (1) whether the modeled shear field globally outperforms random Euler rotations in terms of mean angular misfit, and (2) whether the spatial distribution of misfit exhibits non-random geographic structure. These questions are evaluated using different null models and are not equivalent.

Global Euler-rotation null tests indicate that the mean misfit of the examined shear scenario does not differ significantly from that obtained under random rotations ($p = 0.636$). This result demonstrates that the model does not provide a globally optimal alignment in a scalar, average sense. Importantly, however, this test is insensitive to spatial organization and treats all locations as independent.

In contrast, permutation-based Moran’s I analysis directly evaluates whether misfit values cluster spatially beyond random expectation. This analysis reveals statistically robust spatial autocorrelation from 250 km through at least 3000 km, demonstrating coherent geographic structure in the misfit field. The coexistence of a non-significant global mean test and highly significant spatial autocorrelation indicates that the model captures structured regional and planetary-scale patterns rather than uniformly minimizing misfit everywhere.

Accordingly, the primary empirical result of this study is not improved global alignment, but the detection of long-wavelength spatial organization in stress–misfit patterns that cannot be explained by random spatial processes.

6.8 Scenario Scope and Sensitivity Considerations

The shear field examined in this study corresponds to a single, untuned rotational scenario involving a $\sim 104^\circ$ reorientation along the 31°E meridian. This configuration was selected as a representative test case motivated by prior ECDO-related considerations rather than as an optimized or best-fitting solution.

No attempt has been made here to exhaustively explore parameter space or to identify a globally optimal rotation. Instead, the objective is to assess whether a physically plausible, non-optimized long-wavelength shear scenario produces detectable and statistically robust spatial structure in stress–misfit patterns. The strong spatial

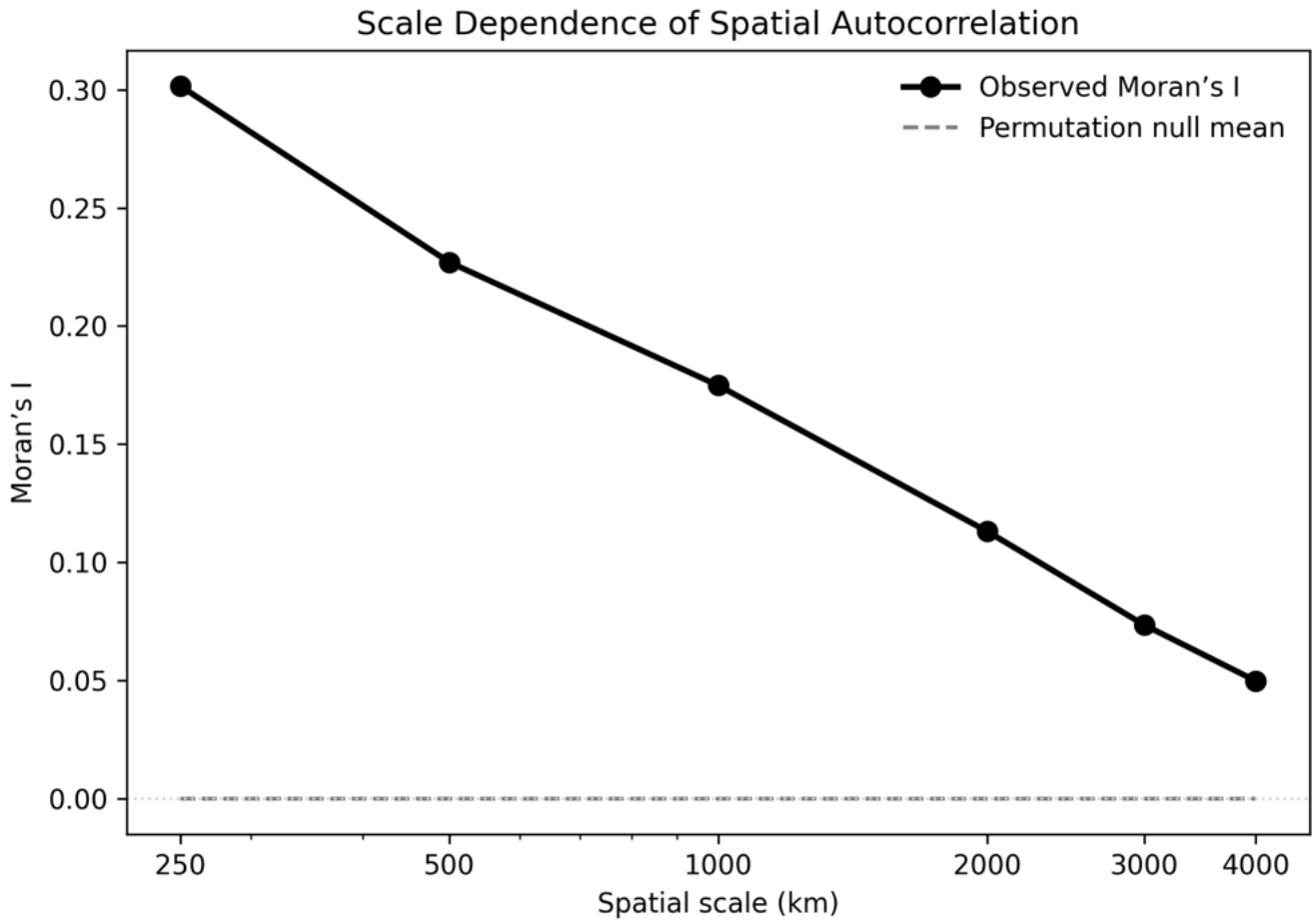


Figure 18: Scale dependence of spatial autocorrelation in the stress-misfit field. Global Moran's I values are shown as a function of characteristic spatial scale from 250 to 4000 km. Observed values (solid symbols) exhibit a smooth, monotonic decay with increasing scale while remaining statistically significant under permutation-based null testing at all examined wavelengths. The permutation null expectation (dashed line) is indistinguishable from zero, indicating that the observed trend reflects genuine spatial organization rather than random spatial structure. The attenuation of Moran's I toward hemispheric scales suggests a finite correlation length and marks the transition from regional clustering to low-degree global modes.

autocorrelation observed across multiple length scales demonstrates that such structure is present even without parameter tuning.

Future work will extend this framework to systematic sensitivity analyses, including alternative rotation axes, magnitudes, and temporal scenarios. The permutation-based Moran’s I methodology introduced here provides a quantitative basis for comparing such scenarios without relying on global mean misfit alone.

6.9 Relation to Paleomagnetic Constraints on True Polar Wander

Proposed true polar wander (TPW) scenarios are subject to constraints derived from paleomagnetic reconstructions, which typically limit long-term TPW amplitudes to tens of degrees over multimillion-year timescales (e.g., Steinberger & Torsvik, 2008). These constraints are primarily sensitive to time-averaged rotational behavior and are optimized to detect sustained, low-frequency reorientations of the solid Earth.

The present analysis does not attempt to reconstruct the timing, duration, or cumulative magnitude of any specific TPW event. Instead, it evaluates whether long-wavelength rotational shear scenarios leave a spatially coherent imprint on present-day stress orientations. As such, the results are compatible with both gradual and episodic reorientation models, including transient or non-steady processes that may not be fully captured by paleomagnetic averaging.

Accordingly, the spatial-statistical findings reported here should be interpreted as evidence for long-wavelength stress organization rather than as direct constraints on TPW kinematics. Integrating spatial stress analyses with paleomagnetic, geodynamic, and stratigraphic constraints represents a key direction for future interdisciplinary work.

6.10 Summary of Empirical Findings

The permutation-based spatial autocorrelation analysis demonstrates that the stress–misfit field is not spatially random at any examined scale. Instead, it exhibits strong and statistically robust geographic organization from local (250 km) through planetary (4000 km) wavelengths. These findings provide quantitative evidence for long-wavelength structure in the stress field and motivate interpretation in terms of global or mantle-coupled forcing mechanisms, which are explored in the Discussion.

7 Results II: Spatial Statistical Structure of the Misfit Field

7.1 Global Mean Alignment and Null-Model Behaviour

Global mean angular misfit between the modeled shear trajectories and WSM stress orientations falls within the empirical distribution generated by Euler-rotated null ensembles. Consistent with earlier findings that global scalar misfit does not, by itself, diagnose model adequacy (e.g. [Cao et al., 2021](#); [Stephan, 2023](#)), the prescribed geometry does not produce a statistically significant reduction in average misfit relative to internally coherent but randomly oriented fields.

This result establishes an important baseline: global mean alignment is non-diagnostic for this geometry and cannot, on its own, be used to justify or reject correspondence between the shear model and observed stress orientations.

7.2 Scale-Dependent Spatial Organisation of the Misfit Field

Permutation-based spatial autocorrelation analysis reveals that the misfit field exhibits statistically significant positive Moran’s I across spatial scales ranging from ~ 250 km to ~ 4000 km. The magnitude of Moran’s I decreases smoothly with increasing scale, ... a behaviour characteristic of genuine spatial correlation rather than sampling artefacts or edge effects in spatially irregular datasets (e.g. [Cliff and Ord, 1981](#); [Fortin and Dale, 2005](#)).

At regional scales (~ 250 – 500 km), clustering reflects coherent misfit domains that coincide with major tectonic regions and plate-boundary systems, consistent with earlier observations that stress orientation varies systematically with tectonic province ([Zoback, 1992](#); [Heidbach et al., 2018](#)). At intermediate scales (~ 500 – 1500 km), autocorrelation remains highly significant, indicating organisation that extends beyond individual tectonic belts.

At long wavelengths (~ 2000 – 4000 km), Moran’s I remains positive and significant, though reduced in magnitude. The persistence of coherence at these scales demonstrates a finite but large correlation length approaching near-hemispheric extents, implying that agreement and disagreement between model and observation are not spatially random, even where global mean alignment remains non-diagnostic.



Figure 19: Geographic distribution of the sampled regions.

7.3 Interpretation of Spatial Organisation in Context

The coexistence of non-diagnostic global mean misfit and statistically significant spatial organisation confirms that these metrics diagnose different aspects of model behaviour (cf. [Stephan, 2023](#)). The spatial results do not imply that the modeled shear field is the dominant or sole control on stress orientation; rather, they indicate that the locations where the model aligns (or misaligns) with observations form geographically structured domains that are unlikely to arise under spatial randomness.

The observed scale-dependent spatial organization is consistent with independent evidence from seismic anisotropy, which demonstrates that upper-mantle deformation beneath continents is coherent, geographically structured, and frequently preserved as fossilized fabric [Silver \(1996\)](#). Taken together, these results suggest that the modeled shear field captures a component of long-wavelength stress organization that is expressed both in present-day stress orientations and in long-lived lithospheric anisotropy. This convergence does not establish causation, but it satisfies a necessary condition of Prediction 8.2: that a persistent shear geometry should leave detectable, spatially organized imprints across independent datasets and observational modalities.

7.4 Geographical Coherence and Cross-Scale Consistency

Low-misfit domains occur along several major tectonic belts and intracontinental regions, while high-misfit do-

main form equally coherent geographic clusters. These domains frequently extend across plate boundaries and, in some cases, across ocean–continent transitions, reflecting organisation at scales larger than individual plate segments. The emergence of such domains parallels earlier observations of regional to continental coherence in stress orientation patterns ([Heidbach et al., 2018](#); [Stephan, 2023](#)), but is here expressed in the misfit field rather than in the stress field itself.

The same areas that exhibit organised misfit structure commonly coincide with regions in which large-scale geological curvature aligns with one of the two modeled shear families or with invariant-contour domains. Although qualitative geometric correspondence and spatial-statistical behaviour are assessed independently, their recurrence in similar regions provides convergent—not causal—support for the hypothesis of long-wavelength structural organisation.

7.5 Comparison with SKS shear-wave splitting observations

We compare the modeled shear-net geometry against published SKS fast-axis orientations compiled by [Silver \(1996\)](#). Each SKS observation is reduced to an axial orientation (0° – 180°) and compared to the local modeled shear azimuth using an axial misfit metric bounded to 0° – 90° . To preserve the spatial structure of the modeled field, statistical significance is assessed using a global-rotation null in which all observed fast-axis orientations are uniformly rotated by a random angle while station locations are held fixed.

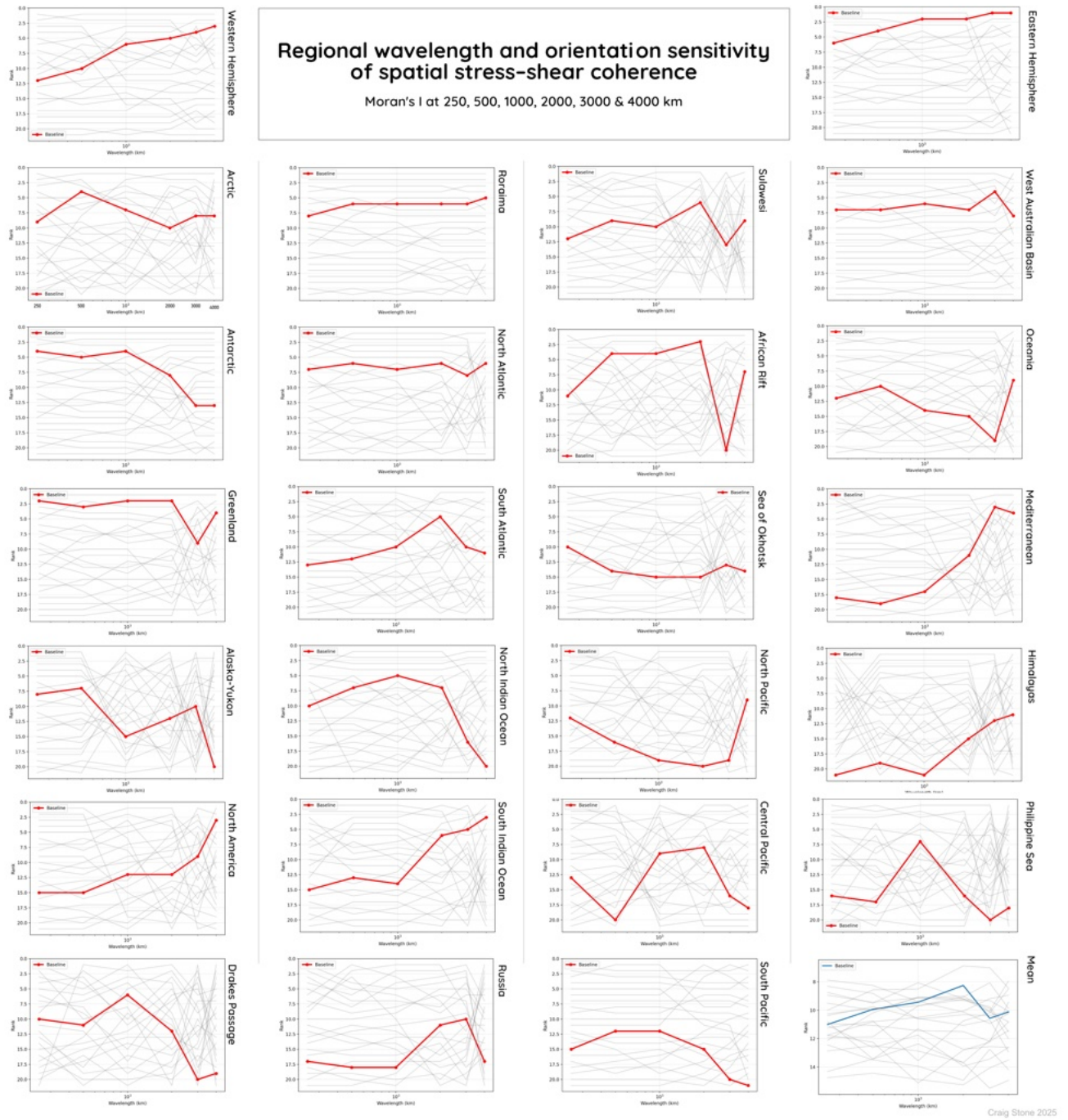
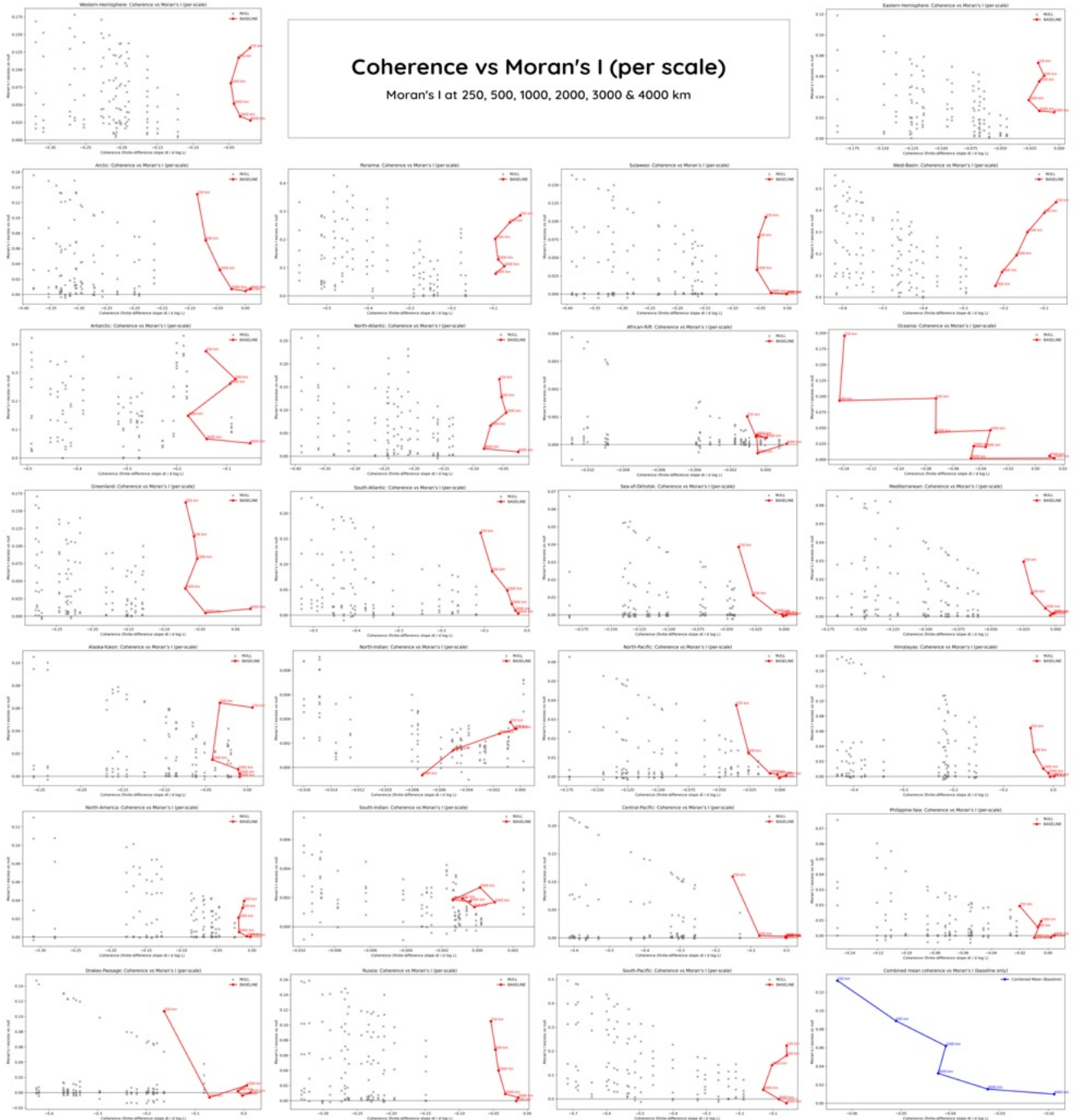


Figure 20: Regional wavelength-orientation sensitivity of spatial stress-shear coherence. For each region, Moran's I was evaluated at 250, 500, 1000, 2000, 3000, and 4000 km wavelengths. The red curve shows the baseline TPW-derived shear geometry; grey curves represent the randomized null shear ensembles. Regions exhibiting plateaued or monotonic trends in the baseline response indicate structured, scale-dependent spatial coherence rather than noise-like behavior.



Craig Stone 2025

Figure 21: Coherence versus Moran's I (per-scale) across regions. Grey points represent the null ensemble; red curves trace the baseline TPW shear field for each wavelength step. The blue curve (lower right) shows the combined mean baseline trajectory relative to the null ensemble. Regions with smooth, structured trajectories show the strongest excess clustering, indicating non-random, scale-consistent spatial organization.

Hemispheric wavelength, orientation and spatial coherence

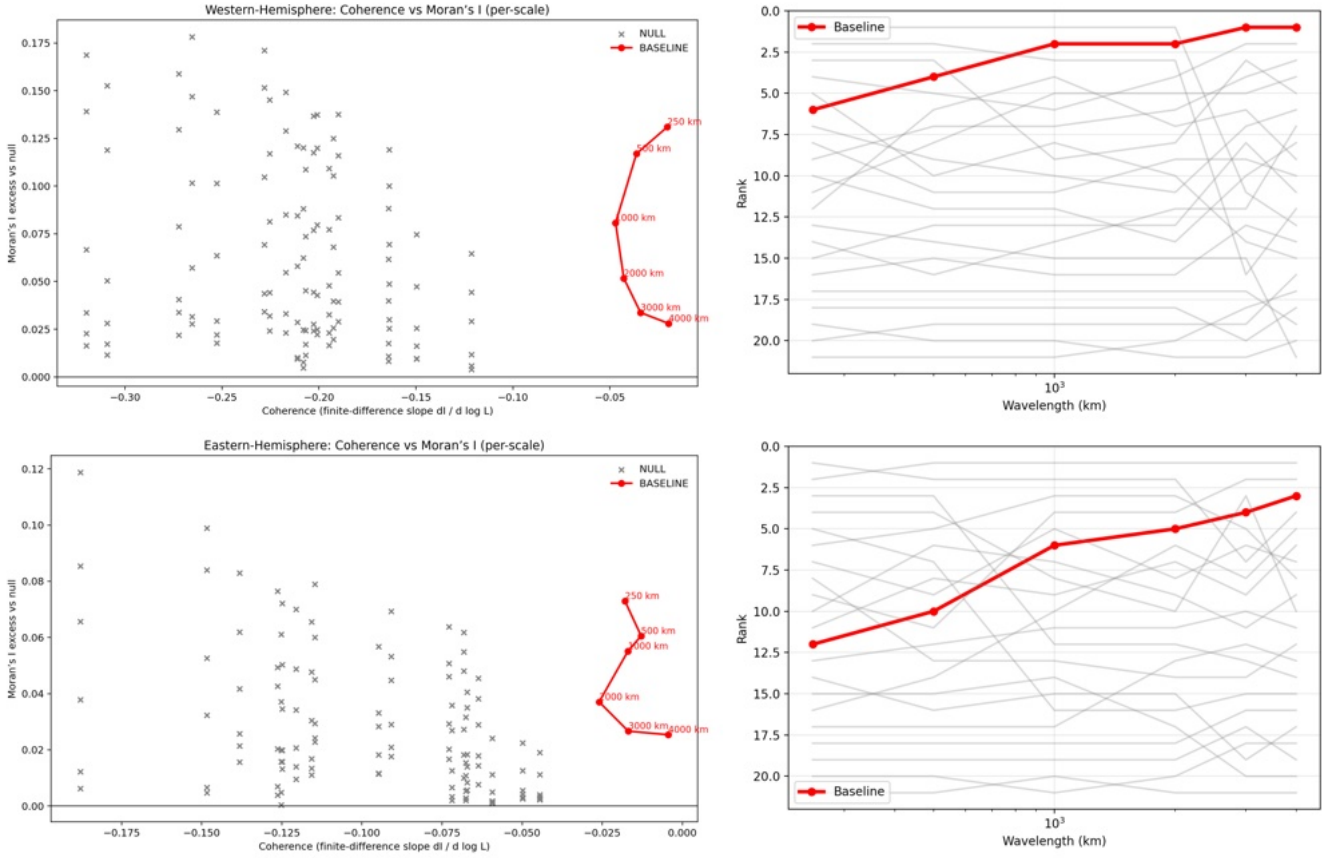


Figure 22: Hemispheric wavelength, orientation, and spatial coherence. The figure illustrates how the spatial organisation of the stress–orientation misfit field varies with wavelength in the Western (top) and Eastern (bottom) hemispheres. In both regions, the baseline shear-net solution (red) occupies a distinct trajectory in coherence–autocorrelation space relative to the rotated-Euler null ensemble (grey), with progressively higher excess Moran’s I at larger wavelengths (left panels). The corresponding rank plots (right panels) show that, across most scales, the baseline solution exhibits stronger spatial organisation of the misfit field than expected under random rotation, particularly at wavelengths of 1000–4000 km. Taken together, these results indicate that the large-scale structure in the misfit field is not a sampling artefact or purely regional effect, but a reproducible hemispheric-scale signal that emerges independently in both datasets. This pattern is consistent with the presence of an organised, long-wavelength component in the underlying stress field.

In the single-family comparison, the observed SKS fast-axis orientations show a systematic shift toward lower misfit angles relative to the null distribution. The observed mean axial misfit ($\sim 24^\circ$) is significantly lower than expected for random axial alignment (45°), with Monte Carlo tests indicating a statistically significant departure from the null for both the mean misfit and the axial variance (Figure 23, Figure 24). This result indicates that the SKS data are non-random with respect to the modeled shear geometry.

Allowing for conjugate shear families further reduces the absolute misfit between observed and modeled orientations; however, when the same minimum-selection procedure is applied to the null ensemble, no additional statistical preference is observed. This behavior is consistent with symmetric finite strain, in which multiple conjugate fabric orientations are geometrically admissible, rather than with a single directed flow alignment.

7.6 Summary of Statistical Findings

The spatial statistical results lead to four principal conclusions:

1. The modeled geometry does not outperform Euler-rotated null ensembles in terms of global mean angular misfit, consistent with prior observations that scalar misfit alone is insufficient to diagnose stress-field correspondence (Cao et al., 2021).
2. The misfit field nevertheless exhibits statistically significant, scale-dependent spatial organisation across regional to near-hemispheric wavelengths under permutation-based testing (following Cliff and Ord, 1981).
3. The smooth decay of Moran's I with increasing scale indicates a finite but large correlation length rather than purely local clustering or random variation.
4. These results are consistent with a hierarchical interpretation in which regional tectonic processes interact with a superimposed long-wavelength organising framework, without implying a unique causal mechanism.

In line with the reviewer's guidance, these findings are interpreted conservatively: the detection of spatial organisation demonstrates that the misfit field is geographically structured and therefore empirically meaningful, but it does not, by itself, identify the physical origin of that structure. The implications of this behaviour are evaluated further in the Discussion and through the falsifiable predictions outlined in Section 8.

7.7 Mid-mantle geometric association between shear-velocity anomalies and Euler domains

To evaluate whether the shear-geometry framework predicts observable structure in independent seismic data, we analysed the spatial distribution of shear-wave velocity anomalies in the global tomography model SEISGLOB2 (Durand et al., 2017). The analysis focused on the mid-mantle depth interval (900–1200 km), where Durand et al. report a marked reorganisation of mantle structure characterised by spectral flattening, slab stagnation, and vertical decorrelation. Two reference depth intervals were also examined for comparison: an upper-mantle band (300–600 km), dominated by plate-coupled dynamics, and a lower-mantle band (1800–2400 km), influenced by long-wavelength thermochemical structure. The SEISGLOB2 model is not used in the construction of the shear-geometry framework and serves exclusively as an external observational test.

Within each depth band, shear-velocity anomalies were selected as grid points exceeding the 90th, 95th, and 97.5th percentiles of $|\ln V_s|$. For each selected anomaly, the angular distance to the nearest of the two Euler domains predicted by the shear-geometry model was computed. These observed distance distributions were compared against two independent null models: (i) a longitude-randomised null preserving latitude and depth, and (ii) a latitude-symmetrised null preserving $|\text{latitude}|$ and depth while randomising longitude. Both nulls retain the depth-dependent sampling density and amplitude distribution of the data while destroying any preferential longitudinal or rotational organisation.

In the mid-mantle depth interval, the cumulative distribution functions (CDFs) of angular distance to the nearest Euler domain show a systematic and statistically robust leftward displacement relative to both null models (Figure 26). This displacement indicates an excess of high-amplitude shear-velocity anomalies occurring at smaller angular distances to Euler domains than expected under randomised geometries. Importantly, the magnitude of the displacement increases monotonically with anomaly percentile: the mean Euler distance is reduced by approximately 1.3° at the 90th percentile, 1.7° at the 95th percentile, and just over 2.0° at the 97.5th percentile relative to the null expectation. The shape of the CDFs further indicates that this excess is distributed across a finite angular basin ($\sim 10\text{--}40^\circ$), rather than concentrated at a singular point.

By contrast, no such Euler-proximal clustering is ob-

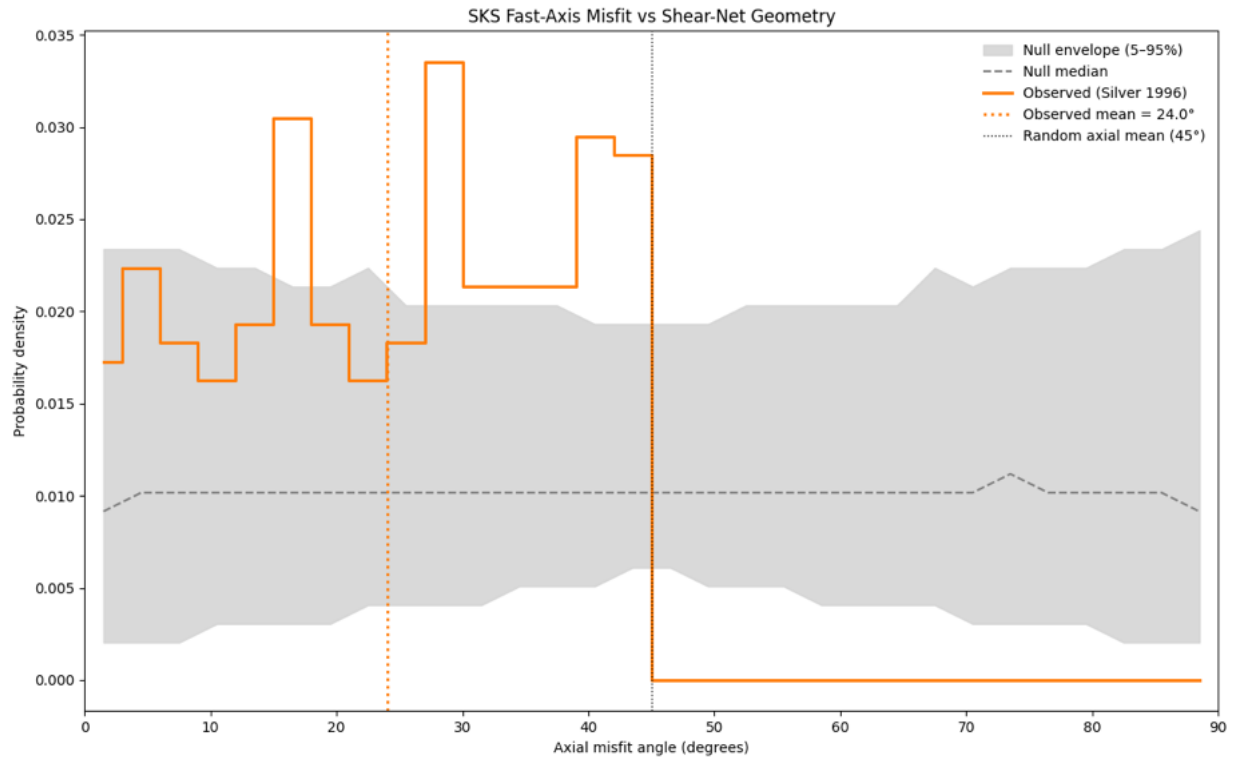


Figure 23: Histogram of axial misfit angles between SKS fast-axis orientations (Silver, 1996) and the modeled shear-net geometry. Misfit angles are bounded to 0° – 90° to account for axial symmetry. The shaded region shows the 5–95% envelope of a global-rotation null distribution, with the dashed line indicating the null median. The observed distribution (orange) is shifted toward lower misfit angles, with a mean misfit of $\sim 24^{\circ}$, significantly below the random axial expectation of 45° .

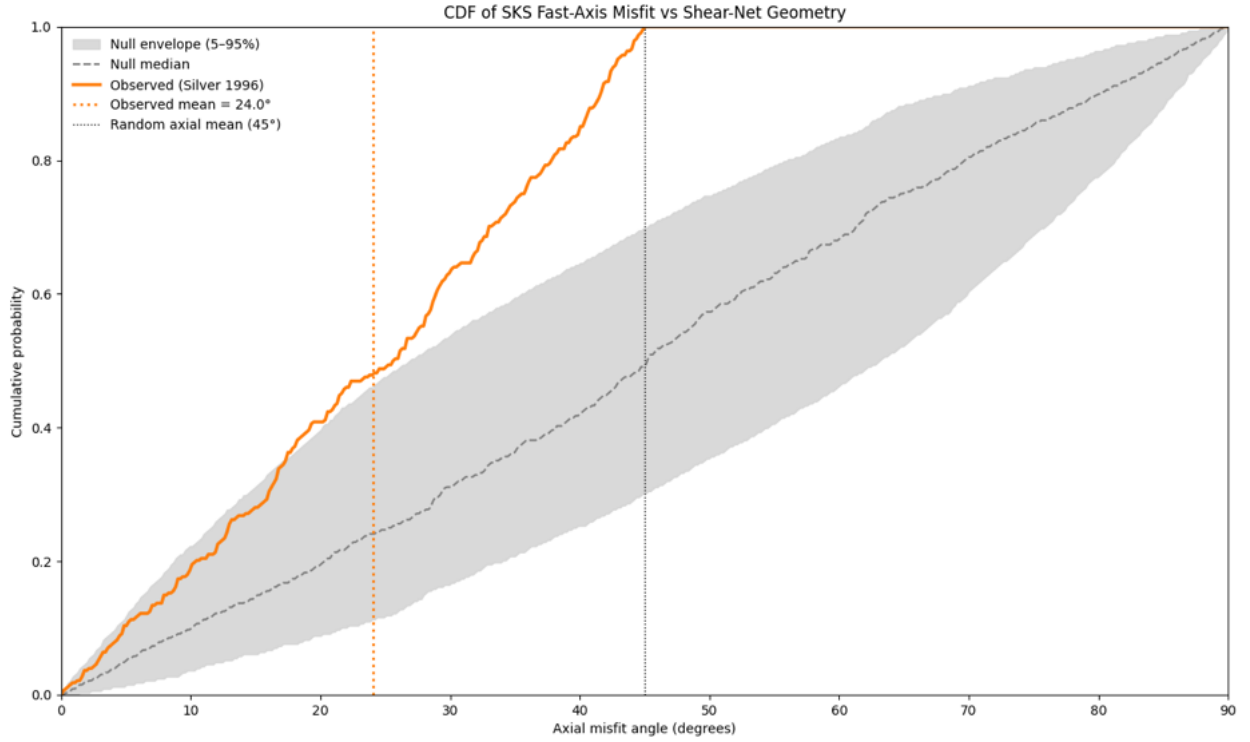


Figure 24: Cumulative distribution function (CDF) of axial misfit angles between SKS fast-axis orientations and the modeled shear-net geometry. The observed CDF (orange) rises more rapidly than the null median, exiting the null envelope at low misfit angles, indicating an excess of well-aligned observations relative to random expectation. This non-parametric representation confirms the statistical significance of the alignment without reliance on binning.

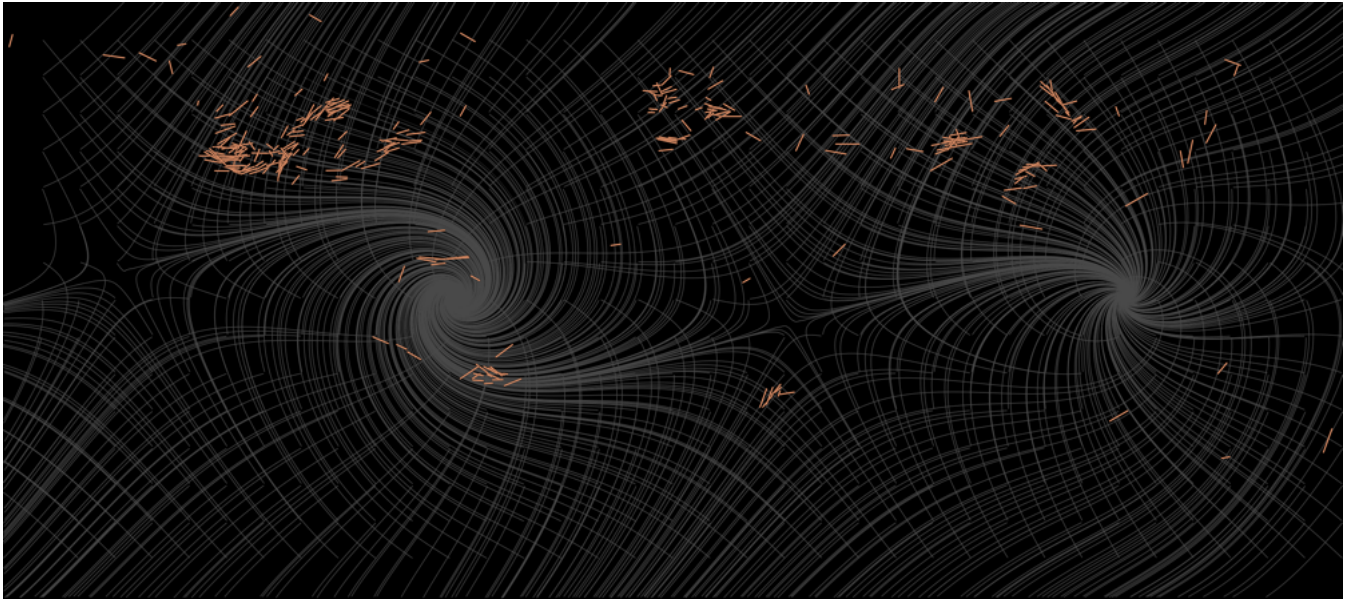


Figure 25: Global spatial comparison between the modeled shear-net geometry (gray streamlines) and SKS fast-axis orientations (orange line segments). The spatial distribution shows that SKS observations preferentially occupy regions of coherent shear while avoiding nodal cores of the modeled field. This figure provides spatial context for the statistical results but is not itself used for inference.

served outside the mid-mantle. In the upper mantle, the distance distributions are systematically shifted toward *larger* angular distances relative to both null models, indicating a repulsion of strong anomalies from Euler domains consistent with plate-dominated deformation. In the lower mantle, the observed distributions are statistically indistinguishable from null expectations, and the effect collapses toward zero as anomaly percentile increases. The opposite signs and depth-localised nature of these effects demonstrate that the mid-mantle clustering is not a coordinate artefact, latitude bias, or trivial consequence of large sample size.

Taken together, these results indicate that strong shear-velocity anomalies in SEISGLOB2 exhibit a depth- and amplitude-dependent geometric association with the Euler domains predicted by the global shear-geometry framework, confined to the mid-mantle depth interval. The coincidence of this association with the independently identified mid-mantle transition in SEISGLOB2 supports interpretation in terms of a kinematic reorganisation of mantle shear, rather than a whole-mantle or plate-controlled process. While this analysis does not establish causation, it demonstrates that the shear-geometry framework makes empirically testable predictions that are satisfied by independent seismic observations.

The confinement of Euler-domain clustering to the mid-mantle depth interval is consistent with interpretations invoking a kinematic reorganisation of mantle shear during episodes of rapid inertial interchange true polar wander (IITPW). In such scenarios, large-scale rotational reorientation of the lithosphere–mantle system would be expected to transmit shear preferentially through depth ranges characterised by reduced vertical coherence and enhanced lateral heterogeneity, rather than uniformly throughout the mantle. The coincidence of Euler-proximal clustering with the mid-mantle transition identified independently in SEISGLOB2 suggests that this depth range may act as a mechanically permissive layer for global-scale shear reorganisation during rapid rotational events, without requiring persistent whole-mantle coupling or long-term plate-driven forcing.

7.8 Implications

Taken together, the statistical outcomes demonstrate that the modeled shear framework is not merely visually compatible with geological curvature, but also exhibits quantifiable spatial organization across scales and regions. The recurring presence of structured wavelength responses and elevated spatial clustering in geologically expressive do-

main is consistent with a planetary-scale stress topology acting in concert with local tectonic and material processes, rather than an artifact of geometric overlay or chance alignment.

8 Testable Predictions

The results presented above demonstrate that the spatial distribution of stress–orientation misfit is non-random and exhibits coherent organisation across multiple spatial scales, even though the modeled shear field does not improve global mean misfit relative to Euler-rotated null ensembles. These findings support the interpretation that a long-wavelength shear topology may act as an organising framework that interacts with, rather than replaces, regional tectonic processes.

However, the reviewer’s concern rightly emphasises that demonstrating spatial organisation is inherently easier than demonstrating causation. To move the framework from interpretive consistency toward empirical evaluation, we articulate here a series of explicit, falsifiable predictions that follow from the hypothesis. These predictions define avenues for independent testing and provide criteria under which the hypothesis may be supported, refined, or rejected.

8.1 Prediction 1: Stability of Misfit Domains Under Dataset Expansion

If the spatial organisation identified in the misfit field reflects a real, long-wavelength component of the stress topology, then coherent misfit domains should remain stable under expansion of the WSM dataset.

Prediction: As additional stress observations are incorporated (either geographically or temporally), the geographic boundaries of major low- and high-misfit domains should remain statistically similar, with domain-scale structure persisting even as local values fluctuate.

Potential falsifier: If the incorporation of new data systematically erodes domain coherence or causes spatial autocorrelation to collapse toward randomness, the inferred long-wavelength organisation would be weakened.

8.2 Prediction 2: Cross-Model Consistency of Spatial Organisation

The hypothesis implies that spatial organisation arises from geometry rather than from a specific stress magnitude distribution.

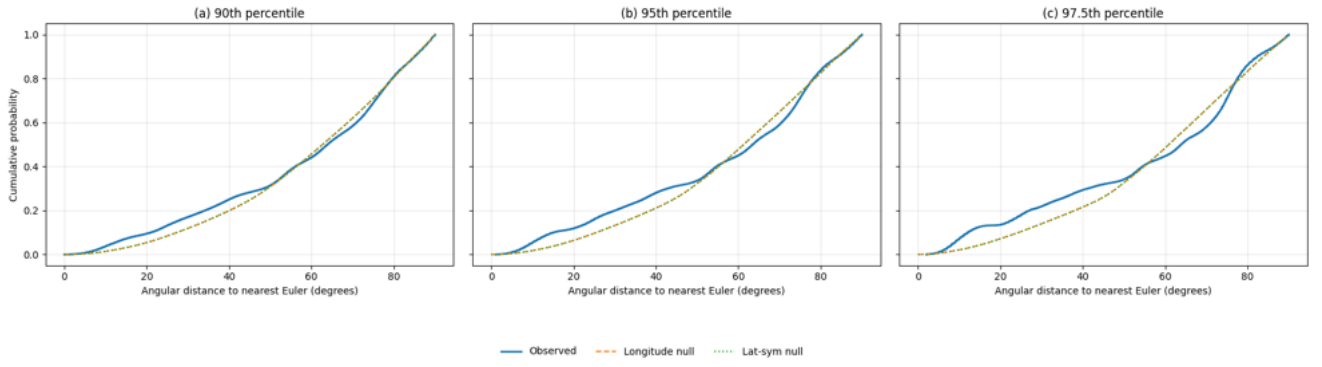


Figure 26: Mid-mantle Euler-domain clustering of shear-velocity extrema in SEISGLOB2. Cumulative distribution functions (CDFs) of angular distance to the nearest Euler domain for high-amplitude mid-mantle (900–1200 km depth) shear-velocity anomalies in the SEISGLOB2 model, shown for the 90th (a), 95th (b), and 97.5th (c) percentiles of $-V_s$. Observed distributions (solid) are compared with longitude-randomised (dashed) and latitude-symmetrised (dotted) null models. In all cases, the observed CDF is systematically shifted toward smaller angular distances relative to both nulls, with the magnitude of the shift increasing monotonically with anomaly amplitude. This depth-localised clustering indicates preferential organisation of strong mid-mantle shear anomalies near the Euler domains predicted by the global shear-geometry framework.

Prediction: When the same statistical analysis is applied to alternative, independently derived global stress models (e.g., numerical or geodynamic models), non-random misfit organisation should recur preferentially in regions predicted to lie near the same shear trajectories or invariant-contour domains.

Potential falsifier: If alternative stress models exhibit spatial organisation in wholly unrelated regions, or if organisation disappears altogether when using independent predictions, the geometric hypothesis would be weakened.

8.3 Prediction 3: Alignment Bias in Newly Forming or Reactivating Structures

If the shear topology reflects a persistent organising framework, then newly forming or reactivated structures should tend, on average, to align with one of the two shear families.

Prediction: Orientation statistics for young fault systems, volcanic chains, and rift segments that postdate major pre-existing fabrics should show directional bias toward Net 1/Net 2 alignment more frequently than expected under random orientation.

Potential falsifier: If young or recently initiated structures exhibit no preferential alignment with the predicted shear trajectories, the inferred organising role would be undermined.

8.4 Prediction 4: Scaling Consistency Across Independent Geometric Proxies

The hypothesis implies that the same long-wavelength organisation should be observable across independent geometric proxies.

Prediction: Regions exhibiting low misfit and shear-aligned curvature in one domain (e.g., rift geometry) should, more often than random expectation, exhibit similar alignment in independent datasets (e.g., gravity-gradient fabrics, bathymetric curvature, glacial excavation patterns).

Potential falsifier: If cross-domain correspondence systematically fails or appears only sporadically without scale consistency, the explanatory value of a persistent shear topology diminishes.

8.5 Prediction 5: Local–Global Interaction Rather Than Global Uniformity

The framework predicts interaction rather than dominance: local tectonics should remain necessary to express deformation.

Prediction: Along-strike variations in curvature and misfit should correlate with transitions in local boundary conditions *while* retaining long-wavelength domain coherence, reflecting superposition rather than replacement of local processes.

Potential falsifier: If curvature and misfit behaviour can be fully explained by local processes alone without residual long-wavelength structure, the added value of the global shear framework would not be supported.

8.6 Prediction 6: Forward-Simulation Replicability

Finally, the hypothesis implies that the observed spatial structure should be reproducible in forward simulations.

Prediction: When simple rheological lithosphere models are subjected to forcing that follows the prescribed shear topology, the resulting deformation fields should reproduce spatially coherent domains comparable in scale and orientation to those observed.

Potential falsifier: If forward models fail to produce domain-scale organisation under a wide range of plausible rheologies and boundary conditions, the geometric hypothesis would require revision.

8.7 Summary and Implications

These predictions do not assume a unique causal mechanism for the inferred shear topology; rather, they define conditions under which the hypothesis gains or loses empirical support. The framework is therefore intentionally conservative: it does not claim causation, but it does make *testable claims* about geometric and statistical behaviour that can be evaluated against new data, independent models, and forward simulations.

9 Synthesis and Discussion

9.1 Relation to Stated Predictions

The results presented here provide an initial empirical evaluation of several stated predictions of the global shear framework. In particular, Prediction 8.2 anticipates that long-wavelength shear organization should be detectable not only in present-day stress data but also in fossilized deformation recorded in the lithosphere–upper mantle system. The statistically significant spatial organization of stress–misfit, together with the independent coherence documented in seismic anisotropy studies, satisfies this prediction at a qualitative and quantitative level, while leaving open questions of mechanism and timing for future work.

9.2 Spatial Organisation Versus Causal Interpretation

The depth-localised Euler-domain clustering observed in SEISGLOB2 coincides with the independently identified mid-mantle transition in shear-velocity structure reported by Durand et al. (2017), providing an observational context for the geometric organisation identified here.

Talbot (2014) demonstrated, at outcrop and shear-zone scale, that curved fabric trajectories preserve a progressive “strain memory” recorded as empirical paths of poles to planes (eppps). These paths provide a robust kinematic reference frame tied to the plane of greatest shear and capture the space–for–time evolution of finite strain across zones of progressive deformation. The present analysis may be viewed as a macro-scale analogue of this behaviour: large-radius curvature systems and invariant–contour alignments exhibit organisation consistent with the same class of inherited, reference-frame-stable deformation trajectories described by Talbot (2014).

The results presented in this study demonstrate that the angular misfit between observed stress orientations and an analytically prescribed global shear field is not spatially random. Instead, the misfit field exhibits statistically significant, scale-dependent spatial organisation from regional to near-hemispheric wavelengths, even though the model does not reduce global mean angular misfit relative to Euler-rotated null ensembles.

This outcome reinforces a central conceptual point: global alignment and spatial organisation diagnose different aspects of model behaviour. The lack of improvement in global mean misfit indicates that the modeled geometry does not act as a globally optimal directional predictor of stress orientation. At the same time, the persistence of spatial autocorrelation shows that the regions of agreement and disagreement are geographically structured in a manner that is unlikely to arise from stochastic variation alone. In other words, the model does not *fit everywhere*, but where it does fit or misfit, those tendencies are spatially organised and scale-consistent.

The combined statistical and spatial analyses indicate that SKS fast-axis orientations are geometrically compatible with the modeled large-scale shear structure. The statistically significant reduction in axial misfit relative to a global-rotation null demonstrates non-random alignment, while the absence of additional preference when conjugate shear families are permitted is consistent with symmetric finite strain rather than directed flow. Together, these results suggest that long-wavelength shear geometry may influence the organization of seismic anisotropy without

uniquely determining fabric orientation at individual sites.

Consistent with the reviewer’s concern, we emphasise that the presence of spatial organisation is not, by itself, evidence of causation. The analyses presented here do not claim to identify a unique physical mechanism, nor do they require that true polar wander be the direct driver of the inferred shear topology. Rather, the results support a more conservative interpretation: that the geometry implied by the prescribed rotational framework captures a long-wavelength organising structure that is detectable in both stress-orientation statistics and the curvature of major geological and geomorphic systems.

9.3 Interaction Between Local Processes and Long-Wavelength Organisation

The regional case studies show that geometric correspondence between the shear trajectories and Earth-surface features is most clearly expressed where the modeled field predicts low-variance shear geometry or convergence between the two shear families. However, in every example, local and regional processes remain necessary to produce the observed structures. Orogenic loading, rifting, slab rollback, glaciation, sediment routing, and mantle upwelling supply the energetic forcing through which deformation is expressed.

Within this framework, the long-wavelength shear topology acts not as a substitute for plate-tectonic or surface processes, but as a geometric constraint that biases the orientation, curvature, and persistence of structures generated by them. This interpretation helps reconcile several recurring observational patterns documented across the examples in this study:

- large-radius curvature that persists across contrasting lithospheric provinces and tectonic episodes;
- directional bimodality consistent with alternating alignment to two conjugate shear families;
- continuity of curvature and segmentation across plate boundaries and ocean–continent transitions; and
- repeated reactivation of shear-aligned corridors through multiple geological cycles.

These behaviours are difficult to explain solely through local kinematics or inheritance, but are consistent with deformation and landscape evolution unfolding within a persistent, planet-scale stress architecture.

9.4 Scale Hierarchy and Correlation Length

The smooth decay of Moran’s I with increasing spatial scale indicates a finite but large correlation length for misfit organisation. Short-wavelength clustering reflects regional tectonics and lithospheric structure, whereas long-wavelength coherence suggests the presence of a low-degree organising mode that spans multiple tectonic domains. The absence of an abrupt transition between these regimes implies hierarchical interaction rather than a dichotomy between local and global control.

This scale-hierarchical behaviour mirrors the qualitative observations: curvature stability, arcuate segmentation, and directional alternation recur at nested wavelengths, from rift segments and continental arcs to ocean-basin fabrics and near-hemispheric structural sweeps. The co-existence of local variability with large-scale coherence is therefore interpreted as a signature of superposition rather than dominance.

9.5 Alternative Explanations and Model Limitations

Several alternative explanations remain plausible and must be considered. Long-wavelength organisation in the misfit field could arise in part from (i) large-scale mantle-flow patterns, (ii) continent-scale rheological contrasts, (iii) plate-boundary network geometry, or (iv) spatial sampling properties of the WSM dataset. Although permutation-based testing mitigates sampling artefacts, the present analysis does not isolate the relative contributions of these processes.

Furthermore, the prescribed rotational geometry represents only one member of a broader class of possible long-wavelength stress fields. Different rotation magnitudes, pole positions, or kinematic formulations may yield alternative shear topologies with similar or improved spatial organisation. The present model should therefore be viewed as a generative hypothesis rather than a unique solution.

Finally, the qualitative comparisons between shear trajectories and geological curvature, while systematic, remain interpretive. Visual congruence does not guarantee mechanical linkage, and future work should replace qualitative comparison with quantitative geometric metrics and forward-model validation.

9.6 Implications for Future Work

The new *Testable Predictions* outlined in Section 8 establish a pathway for advancing the hypothesis from interpretive consistency toward empirical discrimination. In particular, three priorities emerge:

1. **Cross-model evaluation:** repeating the spatial-statistical analysis using independent global stress models to test whether spatial organisation recurs in similar geographic domains.
2. **Temporal and dataset growth tests:** evaluating whether misfit domains remain stable as new stress observations are added or as regional data density increases.
3. **Forward simulations:** examining whether simple rheological lithosphere models subjected to shear-topology forcing reproduce domain-scale spatial structures comparable to those observed.

Together, these avenues convert the present framework from a descriptive alignment exercise into a predictive, falsifiable research programme.

9.7 Synthesis

In synthesis, the results indicate that Earth’s deformation and surface geometry are plausibly organised within a persistent, long-wavelength shear framework that interacts with regional tectonic processes rather than replacing them. The statistical evidence for scale-dependent spatial organisation, combined with recurring geometric correspondence across diverse geological contexts, suggests that the modeled topology captures a meaningful component of Earth’s stress architecture.

At the same time, the interpretation is intentionally conservative: the work demonstrates spatial organisation, not causation. The hypothesis remains open to refinement, rejection, or replacement through the empirical tests identified in Section 8. The value of the framework lies not in asserting a singular mechanism, but in providing a coherent, testable basis for investigating how long-wavelength stress organisation may shape the curvature, segmentation, and spatial hierarchy of geological structures across the planet. In this sense, the framework functions as an auxiliary geometric research program whose value lies in empirical discrimination and predictive extension rather than in claims of mechanistic finality.

10 Conclusions

This study evaluates whether a mathematically prescribed, planet-scale shear topology derived from an idealised TPW-like rotational geometry produces detectable correspondence with observed stress orientations and large-scale geological curvature. The analysis distinguishes explicitly between two diagnostic questions: (i) whether the modeled field improves global mean angular alignment relative to Euler-rotated null ensembles, and (ii) whether the spatial distribution of stress–orientation misfit exhibits statistically significant geographic organisation across multiple spatial scales.

The results show that the modeled geometry does not outperform randomly rotated reference fields in terms of global mean misfit, consistent with earlier findings that scalar misfit statistics alone provide limited diagnostic power when comparing stress models to observations (e.g. [Cao et al., 2021](#); [Stephan, 2023](#)). However, permutation-based spatial autocorrelation analysis demonstrates that the misfit field exhibits significant, scale-dependent spatial organisation from regional to near-hemispheric wavelengths (following [Cliff and Ord, 1981](#)). These findings confirm that global alignment and spatial organisation quantify different aspects of model performance, and that spatial structure may be present even where average misfit remains non-diagnostic.

Qualitative comparison across a wide range of tectonic, geomorphic, and passive-margin systems shows that large-radius curvature, directional bimodality, and structural persistence occur preferentially in regions where the modeled field predicts coherent shear geometry. These correspondences are interpreted as convergent geometric signals rather than causal demonstrations: local and regional processes such as convergence, rifting, magmatism, glaciation, and sediment routing remain essential to the formation and evolution of the examined systems (e.g. [Zoback, 1992](#); [Heidbach et al., 2018](#)). In this framework, the long-wavelength shear topology is understood as an organising environment that interacts with, rather than replaces, plate-tectonic and surface processes.

A key limitation of the present work is that demonstrating spatial organisation is inherently easier than demonstrating causation. The analyses do not identify a unique physical mechanism for the inferred shear topology and do not require that true polar wander be the sole or necessary driver. The contribution of this study is therefore geometric and statistical rather than mechanistic: it shows that a simple, analytically prescribed shear framework produces empirically testable spatial signals that are detectable in

both stress-orientation statistics and surface curvature.

From a methodological perspective, the framework advanced here is best understood as an auxiliary geometric program operating alongside established plate-tectonic theory rather than as a competing paradigm. It does not challenge the hard core assumptions of plate kinematics, mantle convection, or boundary-driven deformation. Instead, it proposes that certain large-scale regularities in Earth-surface geometry and stress orientation may reflect an additional level of organisation that is not captured by local or plate-boundary formulations alone.

Whether this program proves progressive or degenerative will depend on its ability to generate new, independently testable predictions, to delimit conditions under which correspondence should fail, and to integrate with forward geodynamic modelling. The present study establishes only that such a program is empirically admissible and statistically testable.

The *Testable Predictions* outlined in Section 8 define a pathway by which the hypothesis advanced here may be strengthened, refined, or rejected. In particular, future work should (i) evaluate whether spatial organisation persists when applied to independent global stress models, (ii) test the stability of misfit domains as the WSM dataset expands, and (iii) assess whether forward geodynamic simulations subjected to similar long-wavelength forcing reproduce domain-scale spatial structures comparable to those observed. These steps will enable the transition from interpretive consistency toward empirical model discrimination.

In summary, the results support the interpretation that Earth’s deformation and surface geometry may be influenced by a persistent, long-wavelength stress topology that interacts with regional tectonic processes across a hierarchy of spatial scales. While the physical origin of this topology remains open, the evidence presented here demonstrates that its geometric expression is statistically resolvable, internally consistent, and sufficiently structured to justify continued investigation through independent datasets, alternative formulations, and predictive testing.

11 Acknowledgments

The author acknowledges the use of several large-language-model tools (including GPT-5.2, Grok-4.1, Nemotron, LLaMA/Scout, and Claude Sonnet 4.5) in supporting roles during the development of this work. These systems assisted with code drafting and debugging, editorial refinement, conceptual clarification, and “red-teaming”

of the analysis and interpretations (i.e., probing assumptions, testing alternative framings, and identifying potential weaknesses). All empirical analyses, statistical procedures, data processing, model implementation, and scientific interpretation were designed, executed, and validated directly by the author, and were not performed autonomously by any AI system.

References

- Vening Meinesz, F. A. (1947). Shear patterns of the Earth’s crust. *Eos, Transactions of the American Geophysical Union*.
- The Ethical Skeptic (2024). *Exothermic Core-Mantle Decoupling Dzhaniybekov Oscillation (ECDO) Hypothesis*. Available at: <https://theethicalskeptic.com/2024/05/12/exothermic-core-mantle-decoupling-dzhanibekov-oscillation-ecdo-hyp>
- Goldreich, P., and Toomre, A. (1969). Some remarks on polar wandering. *Journal of Geophysical Research*, 74, 2555–2567.
- Tsai, V. C., and Stevenson, D. J. (2007). Theoretical constraints on true polar wander: Effects of mantle viscosity structure. *Journal of Geophysical Research*, 112, B05415.
- Turcotte, D. L., and Schubert, G. (2014). *Geodynamics* (3rd ed.). Cambridge University Press.
- Condie, K. C. (1997). *Plate Tectonics and Crustal Evolution* (4th ed.). Butterworth–Heinemann.
- Zoback, M. L., Zoback, M. D., Adams, J., Assumpção, M., Bell, S., Bergman, E. A., et al. (1989). Global patterns of tectonic stress. *Nature*, 341, 291–298.
- Zoback, M. L. (1992). First- and second-order patterns of stress in the lithosphere. *Journal of Geophysical Research*, 97(B8), 11703–11728.
- Heidbach, O., Rajabi, M., Reiter, K., Ziegler, M., and the WSM Team (2018). Crustal stress pattern across scales based on the World Stress Map database release 2016. *Tectonophysics*, 744, 484–498.
- Sperner, B., Müller, B., Heidbach, O., Delvaux, D., and Reinecker, J. (2003). Tectonic stress in the Earth’s crust: Advances in the World Stress Map project. *Geological Society Special Publication*, 212, 101–116.
- Zang, A., and Stephansson, O. (2010). *Stress Field of the Earth’s Crust*. Springer.
- Stephan, T. (2023). Analyzing the horizontal orientation of the crustal stress field. *Scientific Reports*, 13, 15421.
- Delvaux, D., Moeys, R., Stapel, G., Petit, C., Levi, K., Miroshnichenko, A., Ruzhich, V., and San’kov, V. (1995). Paleostress reconstructions and geodynamics of the Baikal region. *Tectonophysics*, 252, 61–101.
- Saintot, A., Angelier, J., Bergerat, F., Bellier, O., Sassi, W., and Pascal, C. (2003). Paleostress field reconstruction and tectonic evolution at the scale of a plate boundary. *Tectonics*, 22(1), 1007.

- Delvaux, D., and Barth, A. (2012). Paleostress reconstruction and active stress fields in intracontinental settings. *Journal of Structural Geology*, 43, 33–51.
- Ali, S. M., Mahajan, S., Rastogi, B., and Singh, A. K. (2021). Tectonic stress regime and stress patterns from the inversion of earthquake focal mechanisms. *Journal of Geodynamics*, 146, 101865.
- Ping, G., Wang, Y., Li, S., and Zhang, F. (2022). Slip-tendency-based stress inversion from 3D seismic reflection data. *Frontiers in Earth Science*, 10, 812874.
- Liu, C., Hu, J., Chen, Y., and Zhang, H. (2023). Present-day stress field constraints in subduction-zone environments from focal mechanisms. *Frontiers in Earth Science*, 10, 1017632.
- Cao, Z., Li, Q., Liu, M., and Zhu, S. (2021). Origin of three-dimensional crustal stress over the globe: 3-D finite-element modelling. *Journal of Geophysical Research: Solid Earth*, 126(11), e2021JB022137.
- Bada, G., Cloetingh, S., Gerner, P., and Horváth, F. (2007). Present-day stress field and tectonic inversion in the Pannonian Basin. *Tectonophysics*, 410, 1–32.
- Palano, M., Aífa, T., Greco, F., Imprescia, P., and D’Agostino, N. (2015). On the present-day crustal stress–strain-rate fields and mantle anisotropy pattern of Italy. *Geophysical Journal International*, 200, 699–718.
- Ebinger, C. J. (1989). Tectonic development of the western branch of the East African Rift. *Geological Society of America Bulletin*, 101, 885–903.
- Taylor, B., Ebinger, C., d’Almeida, R., et al. (2013). The structure and evolution of the East African Rift System. *Tectonics*, 32, 1–17.
- Corti, G. (2009). Continental rift evolution: From rift initiation to incipient break-up in the East African Rift. *Earth-Science Reviews*, 96, 1–53.
- Philippon, M., and Corti, G. (2015). Obliquity along continental rifts and transfer zones. *Earth-Science Reviews*, 141, 97–133.
- Hoffman, P. F. (1988). United plates of America, the birth of Laurentia. *Annual Review of Earth and Planetary Sciences*, 16, 543–603.
- Dyke, A. S., Moore, A., and Robertson, L. (2004). Deglaciation of North America. *Geological Survey of Canada, Open File* 1574.
- Percival, J. A., Cook, F. A., and Clowes, R. M. (2012). Tectonic evolution of the Canadian Shield. *Canadian Journal of Earth Sciences*, 49, 1177–1193.
- Doré, A. G., Lundin, E. R., Jensen, L. N., et al. (2002). Principal tectonic events in the evolution of the northwest European Atlantic margin. *Geological Society Special Publication*, 196, 1–26.
- Eagles, G., and Scott, D. (2004). Plate convergence west of Patagonia and the Antarctic Peninsula since 61 Ma. *Tectonics*, 23, TC1017.
- Buck, W. R. (2015). Rifting processes and passive margin formation. *Geological Society Special Publication*, 413, 1–27.
- Parsons, B., and Sclater, J. G. (1977). An analysis of the variation of ocean floor bathymetry and heat flow with age. *Journal of Geophysical Research*, 82, 803–827.
- Macdonald, K. C. (1984). Mid-ocean ridges: Fine-scale tectonic, volcanic and hydrothermal processes. *Annual Review of Earth and Planetary Sciences*, 12, 359–399.
- Ballantyne, C. K. (2002). The Quaternary glaciation of Scotland. *Quaternary Science Reviews*, 21, 89–136.
- Woodcock, N. H., and Strachan, R. A. (2014). *Geological History of Britain and Ireland*. Wiley-Blackwell.
- Phillips, E., Auton, C., and Rijdsdijk, K. (2016). Reactivation of basement lineaments in glaciated terrains. *Journal of Structural Geology*, 89, 203–219.
- Johnson, S. E. (1997). Reactivation of Precambrian shear zones. *Journal of the Geological Society*, 154, 1–4.
- Cliff, A. D., and Ord, J. K. (1981). *Spatial Processes: Models and Applications*. Pion.
- Fortin, M.-J., and Dale, M. (2005). *Spatial Analysis: A Guide for Ecologists*. Cambridge University Press.
- Hamilton, W. (1979). *Tectonics of the Indonesian Region*. U.S. Geological Survey Professional Paper 1078.
- Hall, R. (2012). Late Jurassic–Cenozoic reconstructions of SE Asia. *Journal of Asian Earth Sciences*, 44, 1–41.
- Rosencrantz, E., Ross, M. I., and Sclater, J. G. (1990). Age and spreading history of the Cayman Trough. *Journal of Geophysical Research*, 95, 15429–15456.
- Stewart, J. H. (1980). Regional tilt patterns of Late Cenozoic basin-range fault blocks, western United States. *Geological Society of America Bulletin*, 91, 460–464.
- Talbot, C. J. (2014). Empirical paths of poles to planes (eppps) constrain the kinematics of geological shear zones. *Journal of Structural Geology*, 66, 309–333.
- Silver, P. G. (1996). Seismic Anisotropy Beneath the Continents. *Annual Review Earth Planetary Science*, 24, 385–432.
- Durand, S., Debayle, E., Ricard, Y., Zanolli, C. & Lambotte, S. (2017). Confirmation of a change in the global shear velocity pattern at around 1000 km depth. *Geophysical Journal International*, 211(3), 1628–1639. doi:[10.1093/gji/ggx405](https://doi.org/10.1093/gji/ggx405).

A Mathematical Appendix

A.1 Global Shear Field Rotation Geometry and Coordinate Transformation

Let (ϕ, λ) denote geographic latitude and longitude on the unit sphere. The prescribed true-polar-wander-like transformation is implemented as a rigid-body rotation through angle Θ about an axis defined by pole coordinates (ϕ_p, λ_p) .

The position vector in Cartesian form is

$$\mathbf{x} = \begin{bmatrix} \cos \phi \cos \lambda \\ \cos \phi \sin \lambda \\ \sin \phi \end{bmatrix}.$$

The rotation axis unit vector is

$$\mathbf{k} = \begin{bmatrix} \cos \phi_p \cos \lambda_p \\ \cos \phi_p \sin \lambda_p \\ \sin \phi_p \end{bmatrix}.$$

The rotated position is obtained using Rodrigues' rotation formula:

$$\mathbf{x}' = \mathbf{x} \cos \Theta + (\mathbf{k} \times \mathbf{x}) \sin \Theta + \mathbf{k}(\mathbf{k} \cdot \mathbf{x})(1 - \cos \Theta).$$

A.2 Surface Velocity and Shear Directions

The instantaneous surface velocity associated with the rotation is

$$\mathbf{v} = \boldsymbol{\Omega} \times \mathbf{x}, \quad \boldsymbol{\Omega} = \Theta \mathbf{k},$$

which lies tangential to the sphere.

Projecting into the local tangent basis $\{\hat{\mathbf{e}}_\phi, \hat{\mathbf{e}}_\lambda\}$ gives

$$v_\phi = \mathbf{v} \cdot \hat{\mathbf{e}}_\phi, \quad v_\lambda = \mathbf{v} \cdot \hat{\mathbf{e}}_\lambda.$$

The local velocity-gradient tensor on the surface is

$$\mathbf{L} = \begin{bmatrix} \partial v_\phi / \partial \phi & \partial v_\phi / \partial \lambda \\ \partial v_\lambda / \partial \phi & \partial v_\lambda / \partial \lambda \end{bmatrix}, \quad \mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T),$$

where \mathbf{D} is the symmetric strain-rate tensor.

A.3 Principal Shear Trajectories

The principal directions of \mathbf{D} define the conjugate shear families. Let

$$\mathbf{D} \mathbf{e}_i = \sigma_i \mathbf{e}_i, \quad i = 1, 2,$$

with $\sigma_1 \geq \sigma_2$.

Then

- Net 1 trajectories follow the integral curves of \mathbf{e}_1 ,
- Net 2 trajectories follow those of \mathbf{e}_2 .

Invariant-contour loci are defined where the shear differential vanishes:

$$|\sigma_1 - \sigma_2| \rightarrow \min.$$

Trajectories are traced numerically as great-circle-consistent streamlines of \mathbf{e}_i on the sphere

using adaptive step integration of the tangent-plane directions.

A.4 Angular Misfit with WSM Orientations

Given a measured horizontal stress azimuth θ_s and modeled shear direction θ_m at location x , the unsigned angular misfit is

$$\Delta\theta = \min(|\theta_s - \theta_m|, 180^\circ - |\theta_s - \theta_m|),$$

and the field misfit is evaluated as

$$\Delta\theta_{\min} = \min(\Delta\theta_{\text{Net1}}, \Delta\theta_{\text{Net2}}).$$

This quantity forms the scalar field used in the spatial-autocorrelation analysis described in the Methods.

B Source Code

code, figures and data 103.6MB

<https://nobulart.com/media/shear.zip>

shear-map.py

```

1  #!/usr/bin/env python3
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import cartopy.crs as ccrs
5  import cartopy.feature as cfeature
6
7  # -----
8  # 1. Regular lat/lon grid (even spacing!)
9  # -----
10 lon = np.linspace(-180, 180, 721)
11 lat = np.linspace(-80, 80, 321) # Mercator-safe
12 LON, LAT = np.meshgrid(lon, lat)
13
14 lon_r = np.deg2rad(LON)
15 lat_r = np.deg2rad(LAT)
16
17 # -----
18 # 2. TPW specification
19 # -----
20 tpw_lon = np.deg2rad(31.0)
21 tpw_angle = np.deg2rad(104.0)
22
23 # Initial and final pole
24 lat_p0, lon_p0 = np.pi / 2, 0.0
25 lat_p1, lon_p1 = lat_p0 - tpw_angle, tpw_lon
26
27 # -----
28 # 3. Angular distance function
29 # -----
30 def angular_distance(lat_p, lon_p):
31     return np.arccos(
32         np.sin(lat_r) * np.sin(lat_p)

```

```

33         + np.cos(lat_r) * np.cos(lat_p) * np.cos(lon_r
34         ↪ - lon_p)
35     )
36     psi0 = angular_distance(lat_p0, lon_p0)
37     psi1 = angular_distance(lat_p1, lon_p1)
38
39     # -----
40     # 4. Differential centrifugal potential
41     # -----
42     delta_V = np.sin(psi1)**2 - np.sin(psi0)**2
43
44     # -----
45     # 5. Gradients (stress proxy)
46     # -----
47     dV_dlat, dV_dlon = np.gradient(
48         delta_V,
49         np.deg2rad(lat[1] - lat[0]),
50         np.deg2rad(lon[1] - lon[0])
51     )
52
53     theta = np.arctan2(dV_dlon, dV_dlat)
54
55     # Conjugate shear directions
56     shear1 = theta + np.pi / 4
57     shear2 = theta - np.pi / 4
58
59     # Vector components in lat/lon space
60     U1, V1 = np.cos(shear1), np.sin(shear1)
61     U2, V2 = np.cos(shear2), np.sin(shear2)
62
63     # Shear magnitude proxy
64     shear_mag = np.hypot(dV_dlat, dV_dlon)
65
66     # -----
67     # 6. Plot (Mercator via Cartopy)
68     # -----
69     fig = plt.figure(figsize=(16, 8))
70     ax = plt.axes(projection=ccrs.Mercator())
71
72     ax.set_global()
73     ax.coastlines(linewidth=0.8)
74     ax.add_feature(cfeature.BORDERS, linewidth=0.4)
75
76     # Net 1
77     ax.streamplot(
78         lon, lat, U1, V1,
79         transform=ccrs.PlateCarree(),
80         density=1.2,
81         linewidth=0.7,
82         color="black"
83     )
84
85     # Net 2 (conjugate family { thinner, gray)
86     ax.streamplot(
87         lon, lat, U2, V2,
88         transform=ccrs.PlateCarree(),
89         density=1.2,
90         linewidth=0.4,
91         color="dimgray"
92     )
93
94     # Invariant contours (low shear)
95
96     levels = np.percentile(shear_mag, [5, 10])
97     ax.contour(
98         lon, lat, shear_mag,
99         levels=levels,
100        colors="red",
101        linewidths=2,
102        transform=ccrs.PlateCarree()
103    )
104
105    # TPW meridian band
106    for m in [30, 32]:
107        ax.plot(
108            [m, m], [-80, 80],
109            transform=ccrs.PlateCarree(),
110            color="gray",
111            linestyle=":"
112        )
113
114    ax.set_title(
115        "Vening Meinesz{Style Shear Pattern\n"
116        "104° True Polar Wander northward along 30{32°E\n"
117        "Net 1 (solid), Net 2 (dashed), Invariant contours
118        ↪ (red)",
119        fontsize=12
120    )
121    plt.tight_layout()
122    plt.show()

```

shear-kml.py

```

1  #!/usr/bin/env python3
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import simplekml
5
6  # =====
7  ↪ =====
8  # 1. Regular lat/lon grid (Plate Carrée, GIS-safe)
9  # =====
10 ↪ =====
11 lon = np.linspace(-180, 180, 721)
12 lat = np.linspace(-80, 80, 321)
13 LON, LAT = np.meshgrid(lon, lat)
14
15 lon_r = np.deg2rad(LON)
16 lat_r = np.deg2rad(LAT)
17
18 # =====
19 ↪ =====
20 # 2. TPW specification (FIXED: 31°E only)
21 # =====
22 ↪ =====
23 tpw_lon = np.deg2rad(31.0)
24 tpw_angle = np.deg2rad(104.0)
25
26 lat_p0, lon_p0 = np.pi / 2, 0.0
27 lat_p1, lon_p1 = lat_p0 - tpw_angle, tpw_lon
28
29 # =====
30 ↪ =====
31 # 3. Angular distance to pole
32 # =====
33 ↪ =====

```



```

28 def angular_distance(lat_p, lon_p):
29     return np.arccos(
30         np.sin(lat_r) * np.sin(lat_p)
31         + np.cos(lat_r) * np.cos(lat_p) * np.cos(lon_r
    ↳ - lon_p)
32     )
33
34 psi0 = angular_distance(lat_p0, lon_p0)
35 psi1 = angular_distance(lat_p1, lon_p1)
36
37 # =====
38 ↳ =====
39 # 4. Differential centrifugal potential
40 # =====
41 ↳ =====
42 delta_V = np.sin(psi1)**2 - np.sin(psi0)**2
43
44 # =====
45 ↳ =====
46 # 5. Surface gradients (stress proxy)
47 # =====
48 ↳ =====
49 dlat = np.deg2rad(lat[1] - lat[0])
50 dlon = np.deg2rad(lon[1] - lon[0])
51
52 dV_dlat, dV_dlon = np.gradient(delta_V, dlat, dlon)
53 shear_mag = np.hypot(dV_dlat, dV_dlon)
54
55 # =====
56 ↳ =====
57 # 6. Principal stress & conjugate shear directions
58 # =====
59 ↳ =====
60 theta = np.arctan2(dV_dlon, dV_dlat)
61
62 shear1 = theta + np.pi / 4
63 shear2 = theta - np.pi / 4
64
65 U1, V1 = np.cos(shear1), np.sin(shear1)
66 U2, V2 = np.cos(shear2), np.sin(shear2)
67
68 # =====
69 ↳ =====
70 # 7. Invariant (low-shear) contours
71 # =====
72 ↳ =====
73 levels = np.percentile(shear_mag, [5, 10])
74
75 fig, ax = plt.subplots()
76 CS = ax.contour(lon, lat, shear_mag, levels=levels)
77 plt.close(fig)
78
79 # =====
80 ↳ =====
81 # 8. Streamline integrator (explicit geometry)
82 # =====
83 ↳ =====
84 def integrate_streamline(lon0, lat0, U, V, lon, lat,
85     ↳ ds=0.5, nsteps=800):
86     line = []
87     x, y = lon0, lat0
88
89     for _ in range(nsteps):
90         if x < -180 or x > 180 or y < -80 or y > 80:
91             break
92
93     i = np.searchsorted(lon, x) - 1
94     j = np.searchsorted(lat, y) - 1
95
96     if i < 0 or j < 0 or i >= len(lon)-1 or j >=
    ↳ len(lat)-1:
97         break
98
99     u = U[j, i]
100     v = V[j, i]
101     n = np.hypot(u, v)
102
103     if n == 0:
104         break
105
106     x += ds * u / n
107     y += ds * v / n
108     line.append((float(x), float(y)))
109
110 return line
111
112 # =====
113 ↳ =====
114 # 9. Create KML
115 # =====
116 ↳ =====
117 kml = simplekml.Kml()
118
119 # --- Invariant contours ---
120 inv_folder = kml.newfolder(name="Invariant shear
    ↳ contours")
121
122 for i, level in enumerate(CS.levels):
123     for seg in CS.allsegs[i]:
124         if len(seg) < 20:
125             continue
126
127     ls = inv_folder.newlinestring(
128         name=f"Invariant contour ( {level:.2e})"
129     )
130     ls.coords = [(float(x), float(y)) for x, y in
    ↳ seg]
131     ls.style.linestyle.color = simplekml.Color.red
132     ls.style.linestyle.width = 3
133
134 # =====
135 ↳ =====
136 # 10. Net 1 / Net 2 streamlines (50% density reduction)
137 # =====
138 ↳ =====
139 seed_lons = np.arange(-180, 181, 20) # reduced from
    ↳ 10
140 seed_lats = np.arange(-70, 71, 20) # reduced from
    ↳ 10
141
142 net1 = kml.newfolder(name="Net 1 shear trajectories")
143 net2 = kml.newfolder(name="Net 2 shear trajectories")
144
145 for lon0 in seed_lons:
146     for lat0 in seed_lats:
147
148         line1 = integrate_streamline(lon0, lat0, U1,
    ↳ V1, lon, lat)

```

```

134         if len(line1) > 50:
135             ls = net1.newlinestring()
136             ls.coords = line1
137             ls.style.linestyle.color =
138                 ↳ simplekml.Color.black
139             ls.style.linestyle.width = 1
140
141         line2 = integrate_streamline(lon0, lat0, U2,
142                 ↳ V2, lon, lat)
143         if len(line2) > 50:
144             ls = net2.newlinestring()
145             ls.coords = line2
146             ls.style.linestyle.color =
147                 ↳ simplekml.Color.gray
148             ls.style.linestyle.width = 1
149
150         # =====
151         ↳ =====
152         # 11. TPW reference meridian (31°E)
153         # =====
154         ↳ =====
155         ls = kml.newlinestring(name="TPW meridian 31°E")
156         ls.coords = [(31.0, -80.0), (31.0, 80.0)]
157         ls.style.linestyle.color = simplekml.Color.gray
158         ls.style.linestyle.width = 2
159
160         # =====
161         ↳ =====
162         # 12. Save
163         # =====
164         ↳ =====
165         kml.save("meinesz_shear_full_104deg_31E.kml")
166         print("Saved: meinesz_shear_full_104deg_31E.kml")

```

shear-fit.py

```

1  #!/usr/bin/env python3
2  """
3  shear-fit.py
4
5  Compute angular misfit between World Stress Map (WSM)
6  ↳ stress azimuths
7  and two global shear nets. Persist shear geometry and
8  ↳ stress azimuths
9  for downstream Euler-rotation null testing.
10
11  Author: Craig Stone
12  """
13
14  import numpy as np
15  import pandas as pd
16  from pyproj import Geod
17  from scipy.spatial import cKDTree
18  from tqdm import tqdm
19
20  # -----
21  # CONFIGURATION
22  # -----
23
24  WSM_CSV = "WSM_database_2025.csv"
25
26  MAX_MATCH_KM = 200
27  RANDOM_SEED = 42

```

```

27  QUALITY_WEIGHTS = {
28      "A": 1.0,
29      "B": 0.75,
30      "C": 0.4,
31      "D": 0.2
32  }
33
34  # Euler points for diagnostics (example values)
35  EULER_POINTS = {
36      "West": (5.0, -65.0),
37      "East": (-5.0, 120.0)
38  }
39
40  rng = np.random.default_rng(RANDOM_SEED)
41  geod = Geod(ellps="WGS84")
42
43  # -----
44  # UTILITY FUNCTIONS
45  # -----
46
47  def angular_misfit(a, b):
48      d = abs(a - b) % 360
49      d = min(d, 360 - d)
50      return min(d, abs(d - 180))
51
52
53  def angular_distance(lat1, lon1, lat2, lon2):
54      _, _, dist = geod.inv(lon1, lat1, lon2, lat2)
55      return dist / 1000.0 / 111.0
56
57
58  # -----
59  # BUILD SHEAR NETS
60  # -----
61  # NOTE: Replace this section ONLY if you already have
62  # a custom shear-net generator. Otherwise this stub
63  # assumes net midpoints + azimuths are already known.
64
65  def build_shear_net():
66      """
67      Placeholder shear-net generator.
68      Replace with your actual net construction logic.
69      Must return:
70          midpoints: (N, 2) array of (lat, lon)
71          azimuths: (N,) azimuths in degrees
72      """
73      lats = np.linspace(-90, 90, 720)
74      lons = np.linspace(-180, 180, 1440)
75      latg, long = np.meshgrid(lats, lons,
76                 ↳ indexing="ij")
77
78      midpoints = np.column_stack([latg.ravel(),
79                 ↳ long.ravel()])
80      azimuths = (long.ravel() + 90) % 360
81
82      return midpoints, azimuths
83
84  print("Building shear nets...")
85  net1_midpoints, az1 = build_shear_net()
86  net2_midpoints, az2 = build_shear_net() # replace if
87                 ↳ different logic
88
89  tree1 = cKDTree(net1_midpoints)

```

```

88 tree2 = cKDTree(net2_midpoints)
89
90 # -----
91 # PERSIST NET 2 GEOMETRY (CRITICAL)
92 # -----
93
94 print("Saving Net 2 geometry for Euler-rotation null
↳ tests...")
95 np.save("net2_midpoints.npy", net2_midpoints)
96 np.save("net2_azimuths.npy", az2)
97
98 # -----
99 # LOAD AND FILTER WSM
100 # -----
101
102 print("Loading WSM database...")
103 wsm = pd.read_csv(WSM_CSV, low_memory=False)
104
105 required = ["LAT", "LON", "AZI", "REGIME", "QUALITY"]
106 for col in required:
107     if col not in wsm.columns:
108         raise RuntimeError(f"Missing required WSM
↳ column: {col}")
109
110 wsm = wsm.dropna(subset=["LAT", "LON", "AZI"])
111 wsm["LAT"] = pd.to_numeric(wsm["LAT"],
↳ errors="coerce")
112 wsm["LON"] = pd.to_numeric(wsm["LON"],
↳ errors="coerce")
113 wsm["AZI"] = pd.to_numeric(wsm["AZI"],
↳ errors="coerce")
114 wsm = wsm.dropna(subset=["LAT", "LON", "AZI"])
115
116 print("Initial WSM rows:", len(wsm))
117
118 # Optional regime filter (recommended)
119 VALID_REGIMES = {"SS", "TF", "TS"}
120 wsm = wsm[wsm["REGIME"].isin(VALID_REGIMES)]
121
122 print("WSM rows after filtering:", len(wsm))
123
124 # -----
125 # COMPARE TO SHEAR NETS
126 # -----
127
128 results = []
129
130 print("Comparing stress data to shear nets...")
131 for _, row in tqdm(wsm.iterrows(), total=len(wsm)):
132
133     lat = row["LAT"]
134     lon = row["LON"]
135
136     stress_az = row["AZI"] % 360
137     q = row.get("QUALITY", "D")
138     weight = QUALITY_WEIGHTS.get(q, 0.2)
139
140     # Net 1
141     d1, i1 = tree1.query((lat, lon), k=1)
142     if d1 * 111.0 > MAX_MATCH_KM:
143         continue
144     mis1 = angular_misfit(stress_az, az1[i1])
145
146     # Net 2

```

```

147     d2, i2 = tree2.query((lat, lon), k=1)
148     if d2 * 111.0 > MAX_MATCH_KM:
149         continue
150     mis2 = angular_misfit(stress_az, az2[i2])
151
152     entry = {
153         "LAT": lat,
154         "LON": lon,
155         "STRESS_AZ": stress_az,      # <-- REQUIRED FOR
↳ EULER NULL
156         "NET1_MISFIT": mis1,
157         "NET2_MISFIT": mis2,
158         "WEIGHT": weight,
159         "REGIME": row.get("REGIME"),
160         "QUALITY": q,
161         "PLATE": row.get("PLATE", None)
162     }
163
164     for name, (elat, elon) in EULER_POINTS.items():
165         entry[f"DIST_{name}"] = angular_distance(lat,
↳ lon, elat, elon)
166
167     results.append(entry)
168
169 # -----
170 # OUTPUT
171 # -----
172
173 df = pd.DataFrame(results)
174
175 print("\n=== Weighted summary ===")
176 print("Net 1 weighted mean misfit:",
177       np.average(df["NET1_MISFIT"],
↳ weights=df["WEIGHT"]))
178 print("Net 2 weighted mean misfit:",
179       np.average(df["NET2_MISFIT"],
↳ weights=df["WEIGHT"]))
180 print("Fraction where Net 1 fits better:",
181       np.mean(df["NET1_MISFIT"] < df["NET2_MISFIT"]))
182
183 df.to_csv("wsm_shear_misfit_filtered.csv",
↳ index=False)
184 print("Saved wsm_shear_misfit_filtered.csv")

```

shear-euler.py

```

1  #!/usr/bin/env python3
2  """
3  shear_euler.py
4
5  Euler-rotation null test for WSM vs shear-net
↳ alignment.
6  Uses precomputed WSM{shear comparison results.
7
8  Author: Craig Stone
9  """
10
11 import numpy as np
12 import pandas as pd
13 from pyproj import Geod
14 from scipy.spatial import cKDTree
15 from tqdm import tqdm
16
17 # -----

```



```

18 # CONFIGURATION
19 # -----
20
21 CSV_INPUT = "wsm_shear_misfit_filtered.csv"
22
23 # Number of Euler-rotation null realizations
24 N_NULL = 500
25
26 # Max allowed distance (km) for shear matching
27 MAX_MATCH_KM = 200
28
29 # Random seed for reproducibility
30 RANDOM_SEED = 42
31
32 rng = np.random.default_rng(RANDOM_SEED)
33 geod = Geod(ellps="WGS84")
34
35 # -----
36 # LOAD DATA
37 # -----
38
39 print("Loading WSM{shear comparison CSV...}")
40 df = pd.read_csv(CSV_INPUT)
41
42 # Ensure required columns
43 required_cols = ["LAT", "LON", "WEIGHT"]
44 for col in required_cols:
45     if col not in df.columns:
46         raise RuntimeError(f"Missing required column:
↳ {col}")
47
48 # Stress azimuth handling
49 if "STRESS_AZ" not in df.columns:
50     if "AZI" not in df.columns:
51         raise RuntimeError("Neither STRESS_AZ nor AZI
↳ found in CSV")
52     df["STRESS_AZ"] = df["AZI"] % 360
53
54 # Convert to numeric and drop invalid rows
55 for col in ["LAT", "LON", "WEIGHT", "STRESS_AZ"]:
56     df[col] = pd.to_numeric(df[col], errors="coerce")
57
58 df = df.dropna(subset=["LAT", "LON", "WEIGHT",
↳ "STRESS_AZ"])
59
60 print("Rows used for Euler-null test:", len(df))
61
62 # -----
63 # LOAD SHEAR NET MIDPOINTS & AZIMUTHS
64 # (from shear-fit.py output assumptions)
65 # -----
66
67 # These files must be produced during shear-net
↳ construction
68 NET2_MIDPOINTS = "net2_midpoints.npy"
69 NET2_AZIMUTHS = "net2_azimuths.npy"
70
71 mp2 = np.load(NET2_MIDPOINTS) # shape (N, 2) ->
↳ (lat, lon)
72 az2 = np.load(NET2_AZIMUTHS) # shape (N,)
73
74 # -----
75 # EULER ROTATION UTILITIES
76 # -----
77
78 def euler_rotate(lat, lon, pole_lat, pole_lon,
↳ angle_deg):
79     """Rotate a point around an Euler pole."""
80     lat, lon = np.radians(lat), np.radians(lon)
81     pole_lat, pole_lon = np.radians(pole_lat),
↳ np.radians(pole_lon)
82     angle = np.radians(angle_deg)
83
84     r = np.array([
85         np.cos(lat) * np.cos(lon),
86         np.cos(lat) * np.sin(lon),
87         np.sin(lat)
88     ])
89
90     k = np.array([
91         np.cos(pole_lat) * np.cos(pole_lon),
92         np.cos(pole_lat) * np.sin(pole_lon),
93         np.sin(pole_lat)
94     ])
95
96     r_rot = (
97         r * np.cos(angle)
98         + np.cross(k, r) * np.sin(angle)
99         + k * np.dot(k, r) * (1 - np.cos(angle))
100     )
101
102     lat_r = np.degrees(np.arcsin(r_rot[2]))
103     lon_r = np.degrees(np.arctan2(r_rot[1], r_rot[0]))
104
105     return lat_r, lon_r
106
107
108 def rotate_azimuth(lat, lon, az, pole_lat, pole_lon,
↳ angle_deg, dkm=10):
109     """Rotate an azimuth via a short forward step."""
110     lon2, lat2, _ = geod.fwd(lon, lat, az, dkm * 1000)
111     lat_r1, lon_r1 = euler_rotate(lat, lon, pole_lat,
↳ pole_lon, angle_deg)
112     lat_r2, lon_r2 = euler_rotate(lat2, lon2,
↳ pole_lat, pole_lon, angle_deg)
113     az_r, _, _ = geod.inv(lon_r1, lat_r1, lon_r2,
↳ lat_r2)
114     return az_r % 360
115
116
117 def angular_misfit(a, b):
118     d = abs(a - b) % 360
119     d = min(d, 360 - d)
120     return min(d, abs(d - 180))
121
122
123 # -----
124 # BUILD ROTATED SHEAR INDEX
125 # -----
126
127 def build_rotated_shear(mp, az, pole_lat, pole_lon,
↳ angle_deg):
128     rot_pts = []
129     rot_az = []
130
131     for (lat, lon), a in zip(mp, az):
132         lat_r, lon_r = euler_rotate(lat, lon,
↳ pole_lat, pole_lon, angle_deg)

```

```

133     az_r = rotate_azimuth(lat, lon, a, pole_lat,
134     ↪ pole_lon, angle_deg)
135     rot_pts.append((lat_r, lon_r))
136     rot_az.append(az_r)
137
138     rot_pts = np.array(rot_pts)
139     rot_az = np.array(rot_az)
140
141     return cKDTree(rot_pts), rot_az
142
143 # -----
144 # NULL MODEL LOOP
145 # -----
146
147 print(f"Running Euler-rotation null with N =
148 ↪ {N_NULL}")
149
150 null_means = []
151
152 for _ in tqdm(range(N_NULL)):
153     pole_lat = rng.uniform(-90, 90)
154     pole_lon = rng.uniform(-180, 180)
155     angle = rng.uniform(0, 360)
156
157     tree_r, az_r = build_rotated_shear(mp2, az2,
158     ↪ pole_lat, pole_lon, angle)
159
160     misfits = []
161     weights = []
162
163     for _, row in df.iterrows():
164         lat, lon = row["LAT"], row["LON"]
165         stress_az = row["STRESS_AZ"]
166         w = row["WEIGHT"]
167
168         d, i = tree_r.query((lat, lon), k=1)
169         if d * 111 > MAX_MATCH_KM:
170             continue
171
172         m = angular_misfit(stress_az, az_r[i])
173         misfits.append(m)
174         weights.append(w)
175
176     if len(weights) > 0:
177         null_means.append(np.average(misfits,
178     ↪ weights=weights))
179
180 null_means = np.array(null_means)
181
182 # -----
183 # SUMMARY
184 # -----
185
186 observed = np.average(
187     np.minimum(df["NET1_MISFIT"], df["NET2_MISFIT"]),
188     weights=df["WEIGHT"]
189 )
190
191 print("\n=== Euler-rotation null summary ===")
192 print("Observed mean misfit:", observed)
193 print("Null mean:", null_means.mean())
194 print("Null std:", null_means.std())
195 print("p-value:", np.mean(null_means <= observed))

```

```

193
194 # -----
195 # SAVE ONE REPRESENTATIVE NULL MAP
196 # -----
197
198 print("Saving representative null misfit column...")
199
200 pole_lat, pole_lon, angle = 30, -120, 137
201 tree_r, az_r = build_rotated_shear(mp2, az2, pole_lat,
202     ↪ pole_lon, angle)
203
204 null_mis = []
205
206 for _, row in df.iterrows():
207     lat, lon = row["LAT"], row["LON"]
208     stress_az = row["STRESS_AZ"]
209
210     d, i = tree_r.query((lat, lon), k=1)
211     if d * 111 > MAX_MATCH_KM:
212         null_mis.append(np.nan)
213     else:
214         null_mis.append(angular_misfit(stress_az,
215     ↪ az_r[i]))
216
217 df["NULL_MISFIT"] = null_mis
218 df.to_csv("wsm_shear_with_null.csv", index=False)
219
220 print("Saved wsm_shear_with_null.csv")

```

shear-sac-permutation.py

```

1  #!/usr/bin/env python3
2  """
3  shear-sac-permutation.py
4
5  Permutation-based Moran's I spatial autocorrelation
6  ↪ test
7  for WSM{shear misfit fields.
8
9  Correct null model for scalar spatial statistics.
10
11 Author: Craig Stone
12 """
13
14 import numpy as np
15 import pandas as pd
16 from pyproj import Geod
17 from scipy.spatial import cKDTree
18 from tqdm import tqdm
19
20 # -----
21 # CONFIGURATION
22 # -----
23
24 CSV = "wsm_shear_with_null.csv"
25
26 L_SCALES = [250, 500, 1000, 2000, 3000, 4000] # km
27 D_MAX_FACTOR = 3.0
28 N_PERM = 1000 # permutations per
29     ↪ scale
30
31 # -----
32 # LOAD DATA
33 # -----

```

```

32
33 df = pd.read_csv(CSV)
34
35 for col in ["LAT", "LON", "NET1_MISFIT",
36 ↪ "NET2_MISFIT"]:
37     df[col] = pd.to_numeric(df[col], errors="coerce")
38
39 df = df.dropna(subset=["LAT", "LON", "NET1_MISFIT",
40 ↪ "NET2_MISFIT"])
41
42 df["OBS_MISFIT"] = np.minimum(df["NET1_MISFIT"],
43 ↪ df["NET2_MISFIT"])
44
45 lat = df["LAT"].values
46 lon = df["LON"].values
47 values_obs = df["OBS_MISFIT"].values
48
49 N = len(df)
50 print(f"Using {N} spatial points")
51
52 # -----
53 # SPATIAL INDEX
54 # -----
55
56 geod = Geod(ellps="WGS84")
57
58 lat_rad = np.radians(lat)
59 lon_rad = np.radians(lon)
60
61 xyz = np.column_stack([
62     np.cos(lat_rad) * np.cos(lon_rad),
63     np.cos(lat_rad) * np.sin(lon_rad),
64     np.sin(lat_rad)
65 ])
66
67 tree = cKDTree(xyz)
68
69 # -----
70 # PRECOMPUTE NEIGHBORS + WEIGHTS
71 # -----
72
73 def precompute_neighbors_weights(L_km):
74     d_max = D_MAX_FACTOR * L_km
75     radius = d_max / 6371.0 # radians
76
77     neighbors = []
78     weights = []
79
80     for i in tqdm(range(N), desc=f"Precomputing
81 ↪ (L={L_km} km)"):
82         idx = tree.query_ball_point(xyz[i], r=radius)
83
84         js = []
85         ws = []
86
87         for j in idx:
88             if i == j:
89                 continue
90             _, _, d = geod.inv(lon[i], lat[i], lon[j],
91 ↪ lat[j])
92             d /= 1000.0
93
94             if d <= d_max:
95                 js.append(j)
96                 ws.append(np.exp(-d / L_km))
97
98     neighbors.append(np.array(js, dtype=np.int32))
99     weights.append(np.array(ws, dtype=np.float32))
100
101     return neighbors, weights
102
103 # -----
104 # MORAN'S I USING PRECOMPUTED WEIGHTS
105 # -----
106
107 def morans_I(values, neighbors, weights):
108     mean = values.mean()
109     var = np.sum((values - mean) ** 2)
110
111     I_num = 0.0
112     W = 0.0
113
114     for i in range(N):
115         vi = values[i] - mean
116         if vi == 0:
117             continue
118
119         js = neighbors[i]
120         ws = weights[i]
121
122         if len(js) == 0:
123             continue
124
125         vj = values[js] - mean
126         I_num += np.sum(ws * vi * vj)
127         W += np.sum(ws)
128
129     return (N / W) * (I_num / var)
130
131 # -----
132 # MAIN ANALYSIS
133 # -----
134
135 results = []
136
137 print("\nRunning permutation-based Moran's I tests")
138
139 for L in L_SCALES:
140     print(f"\n=== Scale L = {L} km ===")
141
142     neighbors, weights =
143     ↪ precompute_neighbors_weights(L)
144
145     # Observed Moran's I
146     I_obs = morans_I(values_obs, neighbors, weights)
147
148     # Permutation null
149     I_perm = []
150
151     for _ in tqdm(range(N_PERM), desc="Permutations"):
152         permuted = np.random.permutation(values_obs)
153         I_p = morans_I(permuted, neighbors, weights)
154         I_perm.append(I_p)
155
156     I_perm = np.array(I_perm)
157
158     z = (I_obs - I_perm.mean()) / I_perm.std()

```



```

154     p = np.mean(I_perm >= I_obs)
155
156     print(f"Observed I: {I_obs:.5f}")
157     print(f"Null mean: {I_perm.mean():.5f}")
158     print(f"Null std: {I_perm.std():.5f}")
159     print(f"z-score: {z:.2f}")
160     print(f"p-value: {p:.5f}")
161
162     results.append({
163         "L_km": L,
164         "I_obs": I_obs,
165         "I_null_mean": I_perm.mean(),
166         "I_null_std": I_perm.std(),
167         "z_score": z,
168         "p_value": p
169     })
170
171     # -----
172     # SAVE RESULTS
173     # -----
174
175     out = pd.DataFrame(results)
176     out.to_csv("spatial_autocorrelation_permutation_resul
177     ↪ ts.csv", index=False)
178
179     print("\nSaved
180     ↪ spatial_autocorrelation_permutation_results.csv")

```

shear-morans.py

```

1  #!/usr/bin/env python3
2  """
3  Adaptive, sparse, multiprocessing Morans I for TPW
4  ↪ ensemble stress maps.
5
6  Features:
7  • O(N·k) sparse neighbor formulation
8  • Disk-cached neighbors (per scale)
9  • Ensemble-safe (per RUN_ID)
10 • Adaptive permutation stopping
11 • Multiprocessing (Apple Silicon safe)
12 • Progress indicators
13 • Ensemble ranking + confidence intervals
14 • TPW geometry metadata preserved in outputs
15 """
16 import argparse
17 import os
18 import pickle
19 import numpy as np
20 import pandas as pd
21 from tqdm import tqdm
22 from scipy.spatial import cKDTree
23 import multiprocessing as mp
24
25 EARTH_RADIUS_KM = 6371.0
26
27 # =====
28 ↪ =====
29
30 # Geometry utilities
31 # =====
32 ↪ =====
33
34 def haversine_km(lat1, lon1, lat2, lon2):

```

```

32     lat1, lon1, lat2, lon2 = map(np.radians, [lat1,
33     ↪ lon1, lat2, lon2])
34     dlat = lat2 - lat1
35     dlon = lon2 - lon1
36     a = np.sin(dlat/2)**2 +
37     ↪ np.cos(lat1)*np.cos(lat2)*np.sin(dlon/2)**2
38     return 2 * EARTH_RADIUS_KM * np.arcsin(np.sqrt(a))
39
40 def build_neighbors(lat, lon, L_km):
41     coords = np.column_stack((lat, lon))
42     tree = cKDTree(coords)
43     neighbors = []
44
45     for i in tqdm(range(len(lat)),
46     ↪ desc=f"Precomputing neighbors
47     ↪ (L={int(L_km)} km)":
48         idx = tree.query_ball_point(coords[i], r=L_km
49         ↪ / 111.0)
50         idx = [
51             j for j in idx
52             if j != i and haversine_km(lat[i], lon[i],
53             ↪ lat[j], lon[j]) <= L_km
54         ]
55         neighbors.append(np.array(idx,
56         ↪ dtype=np.int32))
57
58     return neighbors
59
60 # =====
61 ↪ =====
62 # Moran's I (sparse)
63 # =====
64 ↪ =====
65
66 def morans_I_sparse(values, neighbors):
67     v = values - values.mean()
68     num = 0.0
69     wsum = 0
70
71     for i, nbrs in enumerate(neighbors):
72         if len(nbrs) == 0:
73             continue
74         num += v[i] * v[nbrs].sum()
75         wsum += len(nbrs)
76
77     den = np.sum(v * v)
78     return (len(values) / wsum) * (num / den)
79
80 # =====
81 ↪ =====
82 # Multiprocessing worker state
83 # =====
84 ↪ =====
85
86 _WORK_VALUES = None
87 _WORK_NEIGHBORS = None
88
89 def _init_worker(values, neighbors):
90     global _WORK_VALUES, _WORK_NEIGHBORS
91     _WORK_VALUES = values
92     _WORK_NEIGHBORS = neighbors

```

```

86
87
88 def _perm_moran(seed):
89     rng = np.random.default_rng(seed)
90     perm = rng.permutation(_WORK_VALUES)
91     return morans_I_sparse(perm, _WORK_NEIGHBORS)
92
93
94 # =====
95 ↪ =====
96 # Main
97 ↪ =====
98
99 def main():
100     ap = argparse.ArgumentParser()
101     ap.add_argument("--input", required=True,
102                     ↪ help="Ensemble CSV")
103     ap.add_argument("--misfit-column", required=True,
104                     ↪ help="Misfit column name")
105     ap.add_argument("--scales", nargs="+", type=float,
106                     default=[250, 500, 1000, 2000,
107                             ↪ 4000])
108     ap.add_argument("--alpha", type=float,
109                     ↪ default=0.01)
110     ap.add_argument("--min-permutations", type=int,
111                     ↪ default=100)
112     ap.add_argument("--max-permutations", type=int,
113                     ↪ default=5000)
114     ap.add_argument("--workers", type=int, default=12)
115     ap.add_argument("--cache-dir",
116                     ↪ default="neighbor_cache")
117     ap.add_argument("--seed", type=int, default=42)
118     ap.add_argument("--output",
119                     ↪ default="morans_ensemble_results.csv")
120     args = ap.parse_args()
121
122     rng = np.random.default_rng(args.seed)
123     os.makedirs(args.cache_dir, exist_ok=True)
124
125     df = pd.read_csv(args.input)
126
127     # -----
128     ↪ -----
129     # Required columns
130     # -----
131     ↪ -----
132
133     required = {"RUN_ID", "LAT", "LON",
134                 ↪ args.misfit_column}
135     missing = required - set(df.columns)
136     if missing:
137         raise RuntimeError(f"Missing required columns:
138                             ↪ {missing}")
139
140     # -----
141     ↪ -----
142     # Extract TPW metadata per RUN_ID
143     # -----
144     ↪ -----
145
146     TPW_META_FIELDS = ["POLE_LAT", "POLE_LON",
147                         ↪ "ROT_DEG", "ROT_SIGN"]
148
149     missing_meta = set(TPW_META_FIELDS) -
150     ↪ set(df.columns)
151     if missing_meta:
152         raise RuntimeError(f"Missing TPW metadata
153                             ↪ columns: {missing_meta}")
154
155     tpw_meta = (
156         df[["RUN_ID"] + TPW_META_FIELDS]
157         .drop_duplicates(subset="RUN_ID")
158         .set_index("RUN_ID")
159     )
160
161     # -----
162     ↪ -----
163     # Geometry (identical across ensemble)
164     # -----
165     ↪ -----
166
167     first_run = df["RUN_ID"].iloc[0]
168     ref = df[df["RUN_ID"] == first_run]
169     lat = ref["LAT"].values
170     lon = ref["LON"].values
171
172     # -----
173     ↪ -----
174     # Neighbor caching
175     # -----
176     ↪ -----
177
178     neighbor_cache = {}
179     for L in args.scales:
180         cache_file = os.path.join(args.cache_dir,
181                                   ↪ f"neighbors_L{int(L)}.pkl")
182         if os.path.exists(cache_file):
183             print(f>Loading cached neighbors for
184                     ↪ L={L} km")
185             with open(cache_file, "rb") as f:
186                 neighbor_cache[L] = pickle.load(f)
187         else:
188             nbrs = build_neighbors(lat, lon, L)
189             with open(cache_file, "wb") as f:
190                 pickle.dump(nbrs, f)
191             neighbor_cache[L] = nbrs
192
193     # -----
194     ↪ -----
195     # Moran's I evaluation
196     # -----
197     ↪ -----
198
199     records = []
200     run_groups = list(df.groupby("RUN_ID"))
201     ctx = mp.get_context("spawn")
202
203     for run_id, group in tqdm(run_groups,
204                               desc="Evaluating
205                                   ↪ ensemble members",
206                               unit="run"):
207
208         values = group[args.misfit_column].values
209         meta = tpw_meta.loc[run_id]
210
211         for L in tqdm(args.scales,
212                       desc=f>Scales for {run_id}",

```

```

185         leave=False):
186
187     neighbors = neighbor_cache[L]
188     I_obs = morans_I_sparse(values, neighbors)
189
190     k = 0
191     n = 0
192     null_I = []
193
194     with ctx.Pool(
195         processes=min(mp.cpu_count(),
196             ↪ args.workers),
197         initializer=_init_worker,
198         initargs=(values, neighbors)
199     ) as pool:
200
201         pbar = tqdm(
202             total=args.max_permutations,
203             desc=f"Permutations (L={L} km)",
204             leave=False
205         )
206
207         while n < args.max_permutations:
208
209             batch = min(100,
210                 ↪ args.max_permutations - n)
211             seeds = rng.integers(0, 2**32 - 1,
212                 ↪ size=batch)
213             results = pool.map(_perm_moran,
214                 ↪ seeds)
215
216             for I_null in results:
217                 null_I.append(I_null)
218                 if abs(I_null) >= abs(I_obs):
219                     k += 1
220                     n += 1
221
222             pbar.update(batch)
223
224             if n >= args.min_permutations:
225                 p_hat = (k + 1) / (n + 1)
226                 if p_hat < args.alpha:
227                     break
228
229         pbar.close()
230
231     null_I = np.array(null_I)
232     null_mean = null_I.mean()
233     null_std = null_I.std(ddof=1)
234     z = (I_obs - null_mean) / null_std if
235     ↪ null_std > 0 else np.inf
236     pval = (k + 1) / (n + 1)
237
238     records.append({
239         "RUN_ID": run_id,
240         "POLE_LAT": meta["POLE_LAT"],
241         "POLE_LON": meta["POLE_LON"],
242         "ROT_DEG": meta["ROT_DEG"],
243         "ROT_SIGN": meta["ROT_SIGN"],
244         "SCALE_KM": L,
245         "I_OBS": I_obs,
246         "I_NULL_MEAN": null_mean,
247         "I_NULL_STD": null_std,
248         "Z_SCORE": z,
249
250         "P_VALUE": pval,
251         "PERMUTATIONS_USED": n
252     })
253
254     results = pd.DataFrame(records)
255     results.to_csv(args.output, index=False)
256
257     # -----
258     ↪ -----
259     # Ensemble ranking + confidence intervals
260     # -----
261     ↪ -----
262
263     summary = (
264         results.groupby("RUN_ID")
265         .agg(
266             MEAN_I=("I_OBS", "mean"),
267             MEDIAN_I=("I_OBS", "median"),
268             I_5P=("I_OBS", lambda x: np.percentile(x,
269                 ↪ 5)),
270             I_95P=("I_OBS", lambda x: np.percentile(x,
271                 ↪ 95))
272         )
273         .reset_index()
274         .merge(tpw_meta.reset_index(), on="RUN_ID",
275             ↪ how="left")
276         .sort_values("MEAN_I", ascending=False)
277     )
278
279     summary["RANK"] = np.arange(1, len(summary) + 1)
280     summary.to_csv("morans_ensemble_ranking.csv",
281         ↪ index=False)
282
283     print("\nSaved:")
284     print(" ●", args.output)
285     print(" ● morans_ensemble_ranking.csv")
286
287     if __name__ == "__main__":
288         main()

```

shear-morans-permutations.py

```

1  #!/usr/bin/env python3
2  import argparse
3  import os
4  import pickle
5  import numpy as np
6  import pandas as pd
7  from tqdm import tqdm
8  from math import radians, sin, cos
9  from scipy.spatial import cKDTree
10
11  EARTH_RADIUS_KM = 6371.0
12  MIN_POINTS = 500
13
14  # =====
15  ↪ =====
16  # Geometry utilities
17  # =====
18  ↪ =====
19
20  def latlon_to_unit(lat, lon):
21     lat = np.radians(lat)

```



```

20     lon = np.radians(lon)
21     return np.column_stack((
22         np.cos(lat) * np.cos(lon),
23         np.cos(lat) * np.sin(lon),
24         np.sin(lat)
25     ))
26
27 def chord_radius(L_km):
28     theta = L_km / EARTH_RADIUS_KM
29     return 2.0 * np.sin(theta / 2.0)
30
31 # =====
32 ↪ =====
33 # Neighbor construction (cached)
34 # =====
35 ↪ =====
36
37 def build_neighbors(lat, lon, L_km, cache_file):
38     if os.path.exists(cache_file):
39         with open(cache_file, "rb") as f:
40             return pickle.load(f)
41
42     print(f"Precomputing neighbors (L={L_km} km)")
43     xyz = latlon_to_unit(lat, lon)
44     tree = cKDTree(xyz)
45     r = chord_radius(L_km)
46
47     neighbors = []
48     for i in tqdm(range(len(xyz))):
49         idx = tree.query_ball_point(xyz[i], r)
50         idx = [j for j in idx if j != i]
51         neighbors.append(np.array(idx,
52             ↪ dtype=np.int32))
53
54     with open(cache_file, "wb") as f:
55         pickle.dump(neighbors, f)
56
57     return neighbors
58
59 # =====
60 ↪ =====
61 # Moran's I (sparse)
62 # =====
63 ↪ =====
64
65 def morans_I_sparse(values, neighbors):
66     x = values - values.mean()
67     num = 0.0
68     W = 0
69
70     for i, nbrs in enumerate(neighbors):
71         if len(nbrs) == 0:
72             continue
73         num += x[i] * x[nbrs].sum()
74         W += len(nbrs)
75
76     den = np.sum(x * x)
77     if W == 0 or den == 0:
78         return np.nan
79
80     return (len(values) / W) * (num / den)
81
82 def permutation_null(values, neighbors, n_perm, rng):
83
84     for i in range(n_perm):
85         perm = rng.permutation(values)
86         null[i] = morans_I_sparse(perm, neighbors)
87     return null
88
89 # =====
90 ↪ =====
91 # Region filtering
92 # =====
93 ↪ =====
94
95 def filter_region(df, region):
96     if region["type"] == "global":
97         return df
98
99     lat0, lon0, R = region["lat"], region["lon"],
100     ↪ region["radius_km"]
101
102     d = np.array([
103         EARTH_RADIUS_KM * np.arccos(
104             max(-1.0, min(1.0,
105                 sin(radians(lat)) * sin(radians(lat0))
106                 ↪ +
107                 cos(radians(lat)) * cos(radians(lat0))
108                 ↪ *
109                 cos(radians(lon - lon0))
110             )))
111     ])
112     for lat, lon in zip(df["LAT"], df["LON"]):
113         return df[d <= R]
114
115 # =====
116 ↪ =====
117 # Main
118 # =====
119 ↪ =====
120
121 def main():
122     ap = argparse.ArgumentParser()
123     ap.add_argument("--input", required=True)
124     ap.add_argument("--misfit-column", required=True)
125     ap.add_argument("--scales", required=True)
126     ap.add_argument("--permutations", type=int,
127         ↪ default=1000)
128     ap.add_argument("--region", action="append")
129     ap.add_argument("--output",
130         ↪ default="morans_results.csv")
131     ap.add_argument("--cache-dir",
132         ↪ default="neighbor_cache")
133     ap.add_argument("--seed", type=int, default=42)
134     args = ap.parse_args()
135
136     os.makedirs(args.cache_dir, exist_ok=True)
137     scales = [float(s) for s in
138         ↪ args.scales.split(",")]
139     rng = np.random.default_rng(args.seed)
140
141     df = pd.read_csv(args.input)
142
143     # =====
144     ↪ =====
145     # Euler pole validation / normalization

```

```

131 # ----- 187
132 ↪ ----- 188
133 if {"POLE_LAT", "POLE_LON"}.issubset(df.columns): 189
134     pole_lat_col = "POLE_LAT" 190
135     pole_lon_col = "POLE_LON" 191
136 elif {"EULER_LAT", 192
137 ↪ "EULER_LON"}.issubset(df.columns): 193
138     pole_lat_col = "EULER_LAT" 194
139     pole_lon_col = "EULER_LON" 195
140 else: 196
141     raise RuntimeError( 197
142         "Input CSV must contain POLE_LAT/POLE_LON 198
143         ↪ or EULER_LAT/EULER_LON" 199
144     ) 200
145 if "RUN_ID" not in df.columns: 201
146     df["RUN_ID"] = "DEFAULT" 202
147 pole_lookup = ( 203
148     df.groupby("RUN_ID")[[pole_lat_col, 204
149 ↪ pole_lon_col]] 205
150     .first() 206
151     .rename(columns={ 207
152         pole_lat_col: "POLE_LAT", 208
153         pole_lon_col: "POLE_LON" 209
154     }) 210
155 ) 211
156 # ----- 212
157 ↪ ----- 213
158 # Regions 214
159 # ----- 215
160 ↪ ----- 216
161 regions = [] 217
162 if args.region: 218
163     for r in args.region: 219
164         name, rest = r.split(":") 220
165         lat, lon, rad = map(float, 221
166 ↪ rest.split(",")) 222
167         regions.append({ 223
168             "name": name, 224
169             "type": "circle", 225
170             "lat": lat, 226
171             "lon": lon, 227
172             "radius_km": rad 228
173         }) 229
174 else: 230
175     regions.append({"name": "GLOBAL", "type": 231
176 ↪ "global"}) 232
177 rows = [] 233
178 skipped_regions = set() 234
179 for run_id, df_run in df.groupby("RUN_ID"): 235
180     print(f"\nRUN_ID: {run_id}") 236
181     pole_lat = pole_lookup.loc[run_id, "POLE_LAT"] 237
182     pole_lon = pole_lookup.loc[run_id, "POLE_LON"] 238
183     for region in regions: 239
184         df_r = filter_region(df_run, region) 240
185         if len(df_r) < MIN_POINTS: 241
186             skipped_regions.add(region["name"]) 242
187         continue
188
189 lat = df_r["LAT"].values
190 lon = df_r["LON"].values
191 vals = df_r[args.misfit_column].values
192
193 for L in scales:
194     cache_file = os.path.join(
195         args.cache_dir,
196         f"neighbors_{region['name']}_L{in_}
197 ↪ t(L)}.pkl"
198     )
199
200 neighbors = build_neighbors(lat, lon,
201 ↪ L, cache_file)
202
203 I_obs = morans_I_sparse(vals,
204 ↪ neighbors)
205 if not np.isfinite(I_obs):
206     skipped_regions.add(region["name"]
207 ↪ )
208     continue
209
210 null = permutation_null(vals,
211 ↪ neighbors, args.permutations, rng)
212
213 rows.append({
214     "RUN_ID": run_id,
215     "REGION": region["name"],
216     "SCALE_KM": L,
217     "N_POINTS": len(vals),
218     "I_OBS": I_obs,
219     "I_NULL_MEAN": null.mean(),
220     "I_NULL_STD": null.std(),
221     "Z_SCORE": (I_obs - null.mean()) /
222 ↪ null.std(),
223     "P_VALUE": (null >= I_obs).mean(),
224     "I_P05": np.percentile(null, 5),
225     "I_P50": np.percentile(null, 50),
226     "I_P95": np.percentile(null, 95),
227     "POLE_LAT": pole_lat,
228     "POLE_LON": pole_lon
229 })
230
231 if not rows:
232     raise RuntimeError("No valid Moran results
233 ↪ were produced for any region.")
234
235 results = pd.DataFrame(rows)
236
237 if "REGION" not in results.columns:
238     raise RuntimeError(
239         "Internal error: REGION column missing
240 ↪ from results."
241     )
242
243 results.to_csv(args.output, index=False)
244 print(f"\nSaved results → {args.output}")
245
246 rankings = (
247     results
248     .sort_values(["REGION", "SCALE_KM", "I_OBS"],
249         ascending=[True, True, False])
250     .assign(RANK=lambda d:

```

```

243         d.groupby(["REGION",
244                    ↪ "SCALE_KM"]).cumcount() + 1)
245     )
246     rank_file = args.output.replace(".csv",
247     ↪ "_rankings.csv")
248     rankings.to_csv(rank_file, index=False)
249     print(f"Saved rankings → {rank_file}")
250
251     if skipped_regions:
252         print("\nSkipped regions due to insufficient
253         ↪ or invalid data:")
254         for r in sorted(skipped_regions):
255             print(f"    - {r}")
256
257     if __name__ == "__main__":
258         main()

```

shear_generate_ensemble.py

```

1  #!/usr/bin/env python3
2  """
3  Generate an ensemble of conjugate shear stress maps
4  ↪ from TPW-style rotations
5  and sample them against the World Stress Map (WSM).
6
7  Baseline:
8  - Rotation = 104°
9  - Axis = 31°E meridian (northward)
10
11 Randomized alternates:
12 - Random Euler poles
13 - Random rotation magnitudes
14 - Random rotation sense
15 """
16 import argparse
17 import numpy as np
18 import pandas as pd
19
20 EARTH_RADIUS = 6371.0
21 DEG2RAD = np.pi / 180.0
22
23 # -----
24 ↪ -----
25 # Geometry utilities
26 # -----
27 ↪ -----
28
29 def sph_to_cart(lat, lon):
30     lat *= DEG2RAD
31     lon *= DEG2RAD
32     return np.array([
33         np.cos(lat) * np.cos(lon),
34         np.cos(lat) * np.sin(lon),
35         np.sin(lat)
36     ])
37
38 def cart_to_sph(v):
39     x, y, z = v
40     lat = np.arcsin(z)
41     lon = np.arctan2(y, x)
42     return lat / DEG2RAD, lon / DEG2RAD

```

```

42 def rotate_vector(v, axis, angle_deg):
43     theta = angle_deg * DEG2RAD
44     axis = axis / np.linalg.norm(axis)
45     return (
46         v * np.cos(theta)
47         + np.cross(axis, v) * np.sin(theta)
48         + axis * np.dot(axis, v) * (1 - np.cos(theta))
49     )
50
51 # -----
52 ↪ -----
53 # TPW stress model (simplified but physical)
54 # -----
55 ↪ -----
56
57 def compute_shear_azimuth(lat, lon, pole_lat,
58 ↪ pole_lon):
59     """
60     Compute conjugate shear azimuths from TPW geometry.
61     """
62     p = sph_to_cart(lat, lon)
63     pole = sph_to_cart(pole_lat, pole_lon)
64
65     # Tangential velocity direction
66     v = np.cross(pole, p)
67     if np.linalg.norm(v) == 0:
68         return np.nan, np.nan
69
70     v /= np.linalg.norm(v)
71
72     # Project to local tangent plane
73     north = np.array([0, 0, 1])
74     east = np.cross(north, p)
75     east /= np.linalg.norm(east)
76     north = np.cross(p, east)
77
78     ve = np.dot(v, east)
79     vn = np.dot(v, north)
80
81     az = (np.arctan2(ve, vn) / DEG2RAD) % 360.0
82
83     # Conjugate shears ±45°
84     return az, (az + 90.0) % 360.0
85
86 # -----
87 ↪ -----
88 # Main
89 # -----
90 ↪ -----
91
92 def main():
93     ap = argparse.ArgumentParser()
94     ap.add_argument("--wsm", required=True, help="WSM
95     ↪ CSV")
96     ap.add_argument("--output",
97     ↪ default="shear_ensemble.csv")
98     ap.add_argument("--random", type=int, default=0,
99     ↪ help="Number of randomized TPW
100     ↪ realizations")
101     ap.add_argument("--rot-min", type=float,
102     ↪ default=30)
103     ap.add_argument("--rot-max", type=float,
104     ↪ default=150)
105     ap.add_argument("--seed", type=int, default=42)

```



```

96     args = ap.parse_args()
97
98     rng = np.random.default_rng(args.seed)
99     wsm = pd.read_csv(args.wsm)
100
101     results = []
102
103     # -----
104     # Define ensemble
105     # -----
106
107     ensemble = []
108
109     # Baseline TPW (104° along 31°E)
110     ensemble.append({
111         "run_id": "TPW_BASELINE",
112         "pole_lat": 0.0,
113         "pole_lon": 31.0,
114         "rot_deg": 104.0,
115         "rot_sign": +1
116     })
117
118     # Randomized alternatives
119     for i in range(args.random):
120         u = rng.uniform(-1, 1)
121         pole_lat = np.arcsin(u) / DEG2RAD
122         pole_lon = rng.uniform(0, 360)
123         rot_deg = rng.uniform(args.rot_min,
124                               ↪ args.rot_max)
125         rot_sign = rng.choice([-1, +1])
126
127         ensemble.append({
128             "run_id": f"TPW_RANDOM_{i:03d}",
129             "pole_lat": pole_lat,
130             "pole_lon": pole_lon,
131             "rot_deg": rot_deg,
132             "rot_sign": rot_sign
133         })
134
135     # -----
136     # Compute stress for each realization
137     # -----
138
139     for cfg in ensemble:
140         print(f"Generating {cfg['run_id']}")
141
142         pole = sph_to_cart(cfg["pole_lat"],
143                             ↪ cfg["pole_lon"])
144         angle = cfg["rot_sign"] * cfg["rot_deg"]
145
146         for _, row in wsm.iterrows():
147             lat, lon = row["LAT"], row["LON"]
148             wsm_az = row["AZI"] % 360.0
149
150             az1, az2 = compute_shear_azimuth(lat, lon,
151                                             ↪ cfg["pol"]
152                                             ↪ e_la
153                                             ↪ t"],
154
155             if np.isnan(az1):
156                 continue
157
158             mis1 = abs((az1 - wsm_az + 90) % 180 - 90)
159             mis2 = abs((az2 - wsm_az + 90) % 180 - 90)
160
161             results.append({
162                 "RUN_ID": cfg["run_id"],
163                 "POLE_LAT": cfg["pole_lat"],
164                 "POLE_LON": cfg["pole_lon"],
165                 "ROT_DEG": cfg["rot_deg"],
166                 "ROT_SIGN": cfg["rot_sign"],
167                 "LAT": lat,
168                 "LON": lon,
169                 "WSM_AZ": wsm_az,
170                 "NET1_AZ": az1,
171                 "NET2_AZ": az2,
172                 "NET1_MISFIT": mis1,
173                 "NET2_MISFIT": mis2,
174                 "BEST_MISFIT": min(mis1, mis2),
175                 "QUALITY": row.get("QUALITY", None),
176                 "PLATE": row.get("PLATE", None)
177             })
178
179     out = pd.DataFrame(results)
180     out.to_csv(args.output, index=False)
181     print(f"Saved {args.output}")
182
183     # -----
184     if __name__ == "__main__":
185         main()

```

rank-plot.py

```

1  #!/usr/bin/env python3
2  import argparse
3  import pandas as pd
4  import numpy as np
5  import matplotlib.pyplot as plt
6  from math import radians, sin, cos, acos
7
8
9  # =====
10 # Geometry utilities
11 # =====
12
13 def angular_distance_deg(lat1, lon1, lat2, lon2):
14     """Great-circle angular distance in degrees."""
15     lat1, lon1, lat2, lon2 = map(radians, [lat1, lon1,
16                                             ↪ lat2, lon2])
17     dot = (
18         sin(lat1) * sin(lat2)
19         + cos(lat1) * cos(lat2) * cos(lon1 - lon2)
20     )
21     dot = max(-1.0, min(1.0, dot))
22     return np.degrees(acos(dot))
23

```

```

24 def euler_polar_coords(lat, lon, lat0, lon0):
25     """
26     Convert Euler pole (lat,lon) into polar coordinates
27     relative to baseline pole (lat0,lon0).
28     """
29     d = angular_distance_deg(lat, lon, lat0, lon0)
30     theta = np.radians(lon)
31     r = np.radians(d)
32     return theta, np.degrees(r)
33
34 # =====
35 ↪ =====
36 # Shared plotting routine
37 # =====
38 ↪ =====
39 def make_plots(df, baseline_id, b_lat, b_lon,
40 ↪ output_prefix):
41     # =====
42     ↪ =====
43     # Plot 1: Rank vs Euler distance (faceted by
44     ↪ scale)
45     # =====
46     ↪ =====
47
48     scales = sorted(df["SCALE_KM"].unique())
49     fig, axes = plt.subplots(
50         1, len(scales),
51         figsize=(6 * len(scales), 5),
52         sharey=True
53     )
54
55     if len(scales) == 1:
56         axes = [axes]
57
58     for ax, L in zip(axes, scales):
59         d = df[df["SCALE_KM"] == L]
60         rand = d[d["RUN_ID"] != baseline_id]
61         base = d[d["RUN_ID"] == baseline_id]
62
63         ax.scatter(
64             rand["EULER_DIST_DEG"], rand["RANK"],
65             s=12, alpha=0.6, label="Random"
66         )
67
68         if not base.empty:
69             ax.scatter(
70                 base["EULER_DIST_DEG"], base["RANK"],
71                 s=50, c="red", label="Baseline"
72             )
73
74         ax.set_title(f"Scale = {int(L)} km")
75         ax.set_xlabel("Euler distance to baseline
76 ↪ (deg)")
77         ax.set_ylabel("Rank")
78         ax.invert_yaxis()
79         ax.grid(True, alpha=0.3)
80
81     plt.tight_layout()
82     plt.savefig(f"{output_prefix}_rank_vs_euler.png",
83 ↪ dpi=200)
84     plt.close()
85
86 # =====
87 ↪ =====
88 # Plot 2: Rank vs wavelength
89 # =====
90 ↪ =====
91
92 fig, ax = plt.subplots(figsize=(7, 5))
93
94 for run_id, g in df.groupby("RUN_ID"):
95     if run_id == baseline_id:
96         ax.plot(
97             g["SCALE_KM"], g["RANK"],
98             linewidth=2.5, label="Baseline"
99         )
100     else:
101         ax.plot(
102             g["SCALE_KM"], g["RANK"],
103             color="gray", alpha=0.3
104         )
105
106 ax.set_xlabel("Wavelength (km)")
107 ax.set_ylabel("Rank")
108 ax.set_xscale("log")
109 ax.invert_yaxis()
110 ax.grid(True, alpha=0.3)
111 ax.legend()
112
113 plt.tight_layout()
114 plt.savefig(f"{output_prefix}_rank_vs_scale.png",
115 ↪ dpi=200)
116 plt.close()
117
118 # =====
119 ↪ =====
120 # Plot 3: Polar Euler-space visualization
121 # =====
122 ↪ =====
123
124 fig = plt.figure(figsize=(7, 7))
125 ax = plt.subplot(111, polar=True)
126
127 for run_id, g in df.groupby("RUN_ID"):
128     theta, r = zip(*[
129         euler_polar_coords(
130             row["POLE_LAT"], row["POLE_LON"],
131             b_lat, b_lon
132         )
133         for _, row in g.iterrows()
134     ])
135     ax.plot(theta, r, alpha=0.5)
136
137 ax.set_title("Euler-space scatter (baseline at
138 ↪ origin)")
139 ax.set_rlabel_position(225)
140 ax.grid(True)
141
142 plt.tight_layout()
143 plt.savefig(f"{output_prefix}_euler_polar.png",
144 ↪ dpi=200)
145 plt.close()
146
147 print("Saved plots:")
148 print(f" {output_prefix}_rank_vs_euler.png")
149 print(f" {output_prefix}_rank_vs_scale.png")

```

```

137     print(f" {output_prefix}_euler_polar.png")
138
139
140 # =====
141 ↪ =====
142 # Main
143 # =====
144 ↪ =====
145
146 def main():
147     ap = argparse.ArgumentParser()
148     ap.add_argument("--input", required=True,
149 ↪     help="Rankings CSV")
150     ap.add_argument("--baseline",
151 ↪     default="TPW_BASELINE")
152     ap.add_argument("--region", default="Arctic")
153     ap.add_argument("--output-prefix",
154 ↪     default="morans_rank")
155     ap.add_argument(
156         "--all-regions-mean",
157         action="store_true",
158         help="Also generate plots for the mean across
159 ↪     all regions combined"
160     )
161     args = ap.parse_args()
162
163     df = pd.read_csv(args.input)
164
165     # -----
166     ↪ -----
167     # Euler pole metadata
168     # -----
169     ↪ -----
170     if not {"POLE_LAT",
171 ↪     "POLE_LON"}.issubset(df.columns):
172         raise RuntimeError(
173             "CSV must include POLE_LAT and POLE_LON
174 ↪     for Euler-space plotting"
175         )
176
177     # Baseline pole
178     b = df[df["RUN_ID"] == args.baseline].iloc[0]
179     b_lat, b_lon = b["POLE_LAT"], b["POLE_LON"]
180
181     # Euler angular distance (degrees)
182     df["EULER_DIST_DEG"] = df.apply(
183         lambda r: angular_distance_deg(
184             r["POLE_LAT"], r["POLE_LON"], b_lat, b_lon
185         ),
186         axis=1
187     )
188
189     # Preserve full dataset for optional averaging
190     df_full = df.copy()
191
192     # =====
193     ↪ =====
194     # Region-specific plots (existing behavior)
195     # =====
196     ↪ =====
197
198     df_region = df[df["REGION"] == args.region]
199     if df_region.empty:
200         raise RuntimeError(
201             f"No valid Moran results available for
202 ↪     region '{args.region}'."
203         )
204
205     make_plots(
206         df_region,
207         baseline_id=args.baseline,
208         b_lat=b_lat,
209         b_lon=b_lon,
210         output_prefix=args.output_prefix
211     )
212
213     # =====
214     ↪ =====
215     # Optional: averaged results across ALL regions
216     # =====
217     ↪ =====
218     if args.all_regions_mean:
219         df_mean = (
220             df_full.groupby(["RUN_ID", "SCALE_KM"],
221 ↪             as_index=False)
222             .agg({
223                 "RANK": "mean",
224                 "EULER_DIST_DEG": "mean",
225                 "POLE_LAT": "mean",
226                 "POLE_LON": "mean"
227             })
228         )
229         df_mean["REGION"] = "ALL_MEAN"
230
231     make_plots(
232         df_mean,
233         baseline_id=args.baseline,
234         b_lat=b_lat,
235         b_lon=b_lon,
236         output_prefix=f"{args.output_prefix}_ALLM",
237 ↪     EAN"
238     )
239
240 if __name__ == "__main__":
241     main()

```

coherence.py

```

1  #!/usr/bin/env python3
2  import argparse
3  import numpy as np
4  import pandas as pd
5  from scipy.signal import savgol_filter
6
7  MIN_POINTS = 3 # require 3 wavelength scales
8
9
10 def curvature_metrics(L, I):
11     """Compute slope, curvature, monotonicity, plateau
12     ↪ index, inflections."""
13     L = np.asarray(L, float)
14     I = np.asarray(I, float)
15
16     # normalize wavelength spacing
17     x = (L - L.min()) / (L.max() - L.min() + 1e-12)
18
19     # smoothing only when enough points exist

```

```

19     if len(I) >= 5:
20         I_sm = savgol_filter(I, window_length=5 if
21             ↪ len(I) >= 5 else len(I)|1,
22             polyorder=2,
23             ↪ mode="interp")
24     else:
25         I_sm = I.copy()
26
27     d1 = np.gradient(I_sm, x, edge_order=1)
28     d2 = np.gradient(d1, x, edge_order=1)
29
30     slope_mean = np.nanmean(d1)
31     slope_var = np.nanvar(d1)
32
33     curvature_mean = np.nanmean(d2)
34     curvature_std = np.nanstd(d2)
35
36     # monotonicity: sign consistency in first
37     ↪ derivative
38     sign_changes = np.sum(np.sign(d1[1:]) !=
39     ↪ np.sign(d1[:-1]))
40     monotonicity = 1.0 - sign_changes / max(len(d1) -
41     ↪ 1, 1)
42
43     # plateau index: fraction where |slope| 0
44     plateau = np.mean(np.abs(d1) <
45     ↪ np.percentile(np.abs(d1), 25))
46
47     # inflections = zero crossings of curvature
48     inf1 = np.sum(np.sign(d2[1:]) != np.sign(d2[:-1]))
49
50     return dict(
51         SLOPE_MEAN=slope_mean,
52         SLOPE_VAR=slope_var,
53         CURVATURE_MEAN=curvature_mean,
54         CURVATURE_STD=curvature_std,
55         MONOTONICITY=monotonicity,
56         PLATEAU_INDEX=plateau,
57         INFLECTIONS=inf1,
58         I_RANGE=float(np.nanmax(I) - np.nanmin(I)),
59     )
60
61 def compute_descriptors(df, out_path):
62
63     rows = []
64     grouped = df.groupby(["REGION", "RUN_ID"],
65     ↪ dropna=False)
66
67     for (region, run_id), g in grouped:
68
69         g = g.sort_values("SCALE_KM")
70         L = g["SCALE_KM"].values
71         I = g["I_OBS"].values
72         Z = g["Z_SCORE"].values
73
74         role = "BASELINE" if run_id == "TPW_BASELINE"
75         ↪ else "PEER"
76
77         if len(L) < MIN_POINTS or np.all(np.isnan(I)):
78             rows.append(dict(
79                 REGION=region, RUN_ID=run_id,
80                 ↪ ROLE=role,
81                 STATUS="INSUFFICIENT_POINTS"
82
83             ))
84             continue
85
86     m = curvature_metrics(L, I)
87     rows.append(dict(
88         REGION=region,
89         RUN_ID=run_id,
90         ROLE=role,
91         STATUS="OK",
92         N_SCALES=len(L),
93         **m,
94         I_MEAN=float(np.nanmean(I)),
95         Z_MEAN=float(np.nanmean(Z)),
96     ))
97
98     desc = pd.DataFrame(rows)
99
100     # ---- Baseline vs Ensemble Coherence ----
101     summaries = []
102     for region, g in desc.groupby("REGION"):
103         base = g[g["ROLE"] == "BASELINE"]
104         peers = g[g["ROLE"] == "PEER"]
105
106         if base.empty or peers.empty:
107             continue
108
109         b = base.iloc[0]
110         diffs = peers.copy()
111
112         for col in [
113             "SLOPE_MEAN", "CURVATURE_STD", "MONOTONICIT
114             ↪ Y",
115             "PLATEAU_INDEX", "INFLECTIONS", "I_RANGE"
116         ]:
117             diffs[f"DELTA_{col}"] = peers[col] -
118             ↪ b[col]
119
120         summaries.append(dict(
121             REGION=region,
122             N_PEERS=len(peers),
123             BASE_SLOPE_MEAN=b["SLOPE_MEAN"],
124             ENSEMBLE_SLOPE_MEAN=float(peers["SLOPE_ME
125             ↪ AN"].mean()),
126             DELTA_SLOPE_MEAN=float(diffs["DELTA_SLOPE
127             ↪ _MEAN"].mean()),
128
129             BASE_CURVATURE_STD=b["CURVATURE_STD"],
130             ENSEMBLE_CURVATURE_STD=float(peers["CURVA
131             ↪ TURE_STD"].mean()),
132             DELTA_CURVATURE_STD=float(diffs["DELTA_CU
133             ↪ RVATURE_STD"].mean()),
134
135             BASE_MONOTONICITY=b["MONOTONICITY"],
136             ENSEMBLE_MONOTONICITY=float(peers["MONOTO
137             ↪ NICITY"].mean()),
138             DELTA_MONOTONICITY=float(diffs["DELTA_MON
139             ↪ OTONICITY"].mean()),
140
141             BASE_PLATEAU_INDEX=b["PLATEAU_INDEX"],
142             ENSEMBLE_PLATEAU_INDEX=float(peers["PLATE
143             ↪ AU_INDEX"].mean()),
144             DELTA_PLATEAU_INDEX=float(diffs["DELTA_PL
145             ↪ ATEAU_INDEX"].mean()),
146         ))

```



```

128
129     summary = pd.DataFrame(summaries)
130
131     desc.to_csv(out_path, index=False)
132     summary.to_csv(out_path.replace(".csv",
133     ↪ "_summary.csv"), index=False)
134
135     print(f"Saved descriptors → {out_path}")
136     print(f"Saved ensemble summary
137     ↪ {out_path.replace('.csv', '_summary.csv')}")
138
139 def main():
140     ap = argparse.ArgumentParser()
141     ap.add_argument("--input", required=True)
142     ap.add_argument("--output",
143     ↪ default="coherence_shape_descriptors.csv")
144     args = ap.parse_args()
145
146     df = pd.read_csv(args.input)
147     required = {
148         "RUN_ID", "REGION", "SCALE_KM", "I_OBS", "Z_SCORE"
149     }
150     missing = required - set(df.columns)
151     if missing:
152         raise RuntimeError(f"Missing columns:
153         ↪ {missing}")
154
155     compute_descriptors(df, args.output)
156
157 if __name__ == "__main__":
158     main()

```

coherence_v_{*s*}_{*m*}_{*oran*}.py

```

1  #!/usr/bin/env python3
2  import argparse
3  import os
4  import numpy as np
5  import pandas as pd
6  import matplotlib.pyplot as plt
7
8
9  # -----
10 ↪ -----
11 # Robust finite-difference slope estimator (dI / d log
12 ↪ L)
13 # -----
14 ↪ -----
15
16 def finite_difference_slopes(L, I):
17     """
18     Compute dI/d(log L) with numerical stability.
19     Handles duplicate scales, missing values, and
20     ↪ sparse scale sets.
21     Returns one slope per scale value.
22     """
23     L = np.asarray(L, dtype=float)
24     I = np.asarray(I, dtype=float)
25
26     logL = np.log(L)
27     slopes = np.full_like(I, np.nan)
28
29     n = len(L)

```

```

25     if n < 2:
26         return slopes
27
28     for i in range(n):
29         il = i - 1 if i > 0 else None
30         ir = i + 1 if i < n - 1 else None
31
32         # central difference (preferred)
33         if il is not None and ir is not None:
34             dlog = logL[ir] - logL[il]
35             if dlog != 0:
36                 slopes[i] = (I[ir] - I[il]) / dlog
37                 continue
38
39         # forward fallback
40         if ir is not None:
41             dlog = logL[ir] - logL[i]
42             if dlog != 0:
43                 slopes[i] = (I[ir] - I[i]) / dlog
44                 continue
45
46         # backward fallback
47         if il is not None:
48             dlog = logL[i] - logL[il]
49             if dlog != 0:
50                 slopes[i] = (I[i] - I[il]) / dlog
51                 continue
52
53     return slopes
54
55 # -----
56 ↪ -----
57 # Load and merge coherence + Moran's outputs
58 # -----
59 ↪ -----
60
61 def load_data(coherence_path, morans_path):
62     coh = pd.read_csv(coherence_path)
63     mor = pd.read_csv(morans_path)
64
65     mor = mor.copy()
66     mor["I_EXCESS_MEAN"] = mor["I_OBS"] -
67     ↪ mor["I_NULL_MEAN"]
68
69     df = pd.merge(
70         coh,
71         mor[
72             [
73                 "RUN_ID",
74                 "REGION",
75                 "SCALE_KM",
76                 "I_OBS",
77                 "I_NULL_MEAN",
78                 "I_EXCESS_MEAN",
79             ]
80         ],
81         on=["RUN_ID", "REGION"],
82         how="inner",
83     )
84
85     return df

```

```

85 # ----- 138
86 ↪ ----- 139
87 # Global scatter / baseline highlighted 140
88 # ----- 141
89 ↪ ----- 142
88 def plot_global(df, outdir): 143
89     fig, ax = plt.subplots(figsize=(10, 6)) 144
90 145
91     peers = df[df["ROLE"] != "BASELINE"] 146
92     base = df[df["ROLE"] == "BASELINE"] 147
93 148
94     ax.scatter( 149
95         peers["SLOPE_MEAN"], 150
96         peers["I_EXCESS_MEAN"], 151
97         c="0.5", 152
98         marker="x", 153
99         alpha=1.0, 154
100         s=26, 155
101         label="NULL", 156
102         zorder=1, 157
103     ) 158
104 159
105     ax.scatter( 160
106         base["SLOPE_MEAN"], 161
107         base["I_EXCESS_MEAN"], 162
108         c="red", 163
109         s=50, 164
110         zorder=4, 165
111         label="BASELINE", 166
112     ) 167
113 168
114     ax.axhline(0, color="black", lw=0.8) 169
115     ax.set_title("Global coherence vs spatial 170
116 ↪ clustering strength (per-scale)") 171
117     ax.set_xlabel("Coherence (finite-difference slope 172
118 ↪ dI / d log L)") 173
119     ax.set_ylabel("Morans I excess vs null (I_obs 174
120 ↪ I_null_mean)") 175
121     ax.legend(frameon=False) 176
122 177
123     fig.tight_layout() 178
124     fig.savefig(f"{outdir}/global_coherence_vs_I.png", 179
125 ↪ dpi=220) 180
126     plt.close(fig) 181
127 182
128 # ----- 183
129 ↪ ----- 184
126 # Combined mean baseline trajectory (BLUE LINE) 185
127 # ----- 186
128 ↪ ----- 187
128 def plot_combined_mean(df, outdir): 188
129     base = df[df["ROLE"] == "BASELINE"].copy() 189
130 190
131     mean_curve = ( 191
132         base.groupby("SCALE_KM", 192
133             ↪ as_index=False)[["I_OBS", 193
134             ↪ "I_EXCESS_MEAN"]] 194
135         .mean() 195
136         .sort_values("SCALE_KM") 196
137     ) 197
138 198
139     mean_curve["COHERENCE_SLOPE"] = 199
140     ↪ finite_difference_slopes( 200
141         mean_curve["SCALE_KM"].values, 201
142         mean_curve["I_OBS"].values, 202
143     ) 203
144 204
145     fig, ax = plt.subplots(figsize=(10, 6)) 205
146 206
147     ax.plot( 207
148         mean_curve["COHERENCE_SLOPE"], 208
149         mean_curve["I_EXCESS_MEAN"], 209
150         "-o", 210
151         color="blue", 211
152         lw=2.2, 212
153         ms=6, 213
154         zorder=4, 214
155         label="Combined Mean (Baseline)", 215
156     ) 216
157 217
158     for _, r in mean_curve.iterrows(): 218
159         ax.text( 219
160             r["COHERENCE_SLOPE"], 220
161             r["I_EXCESS_MEAN"], 221
162             f"{int(r['SCALE_KM'])} km", 222
163             fontsize=9, 223
164             color="blue", 224
165             ha="left", 225
166             va="bottom", 226
167         ) 227
168 228
169     ax.axhline(0, color="black", lw=0.8) 229
170     ax.set_title("Combined mean coherence vs Morans I 230
171 ↪ (baseline only)") 231
172     ax.set_xlabel("Coherence (finite-difference slope 232
173 ↪ dI / d log L)") 233
174     ax.set_ylabel("Moran's I excess vs null") 234
175 235
176     ax.legend(frameon=False) 236
177     fig.tight_layout() 237
178     fig.savefig(f"{outdir}/combined_mean_coherence_vs_ 238
179 ↪ _I.png", dpi=220) 239
180     plt.close(fig) 240
181 241
182 # ----- 242
183 ↪ ----- 243
184 # Regional plot / baseline trajectory vs null ensemble 244
185 # ----- 245
186 ↪ ----- 246
187 def plot_region(df_region, outdir): 247
188     region = df_region["REGION"].iloc[0] 248
189 249
190     baseline = ( 250
191         df_region[df_region["ROLE"] == "BASELINE"] 251
192         .sort_values("SCALE_KM") 252
193         .copy() 253
194     ) 254
195 255
196     peers = df_region[df_region["ROLE"] != "BASELINE"] 256
197 257
198     slopes = finite_difference_slopes( 258
199         baseline["SCALE_KM"].values, 259
200         baseline["I_OBS"].values, 260
201     ) 261
202 262
203     baseline["COHERENCE_SLOPE"] = slopes 263
204 264
205     fig, ax = plt.subplots(figsize=(10, 6)) 265

```

```

197
198     ax.scatter(
199         peers["SLOPE_MEAN"],
200         peers["I_EXCESS_MEAN"],
201         s=26,
202         c="0.5",
203         marker="x",
204         alpha=1.0,
205         label="NULL",
206         zorder=1,
207     )
208
209     ax.plot(
210         baseline["COHERENCE_SLOPE"],
211         baseline["I_EXCESS_MEAN"],
212         "-o",
213         color="red",
214         lw=2.0,
215         ms=6,
216         zorder=4,
217         label="BASELINE",
218     )
219
220     for _, r in baseline.iterrows():
221         ax.text(
222             r["COHERENCE_SLOPE"],
223             r["I_EXCESS_MEAN"],
224             f"{int(r['SCALE_KM'])} km",
225             fontsize=9,
226             color="red",
227             ha="left",
228             va="bottom",
229         )
230
231     ax.axhline(0, color="black", lw=0.8)
232     ax.set_title(f"{region}: Coherence vs Morans I
233 ↪ (per-scale)")
234     ax.set_xlabel("Coherence (finite-difference slope
235 ↪ dI / d log L)")
236     ax.set_ylabel("Moran's I excess vs null")
237
238     ax.legend(frameon=False)
239     fig.tight_layout()
240     fig.savefig(f"{outdir}/{region}_coherence_vs_I.pn
241 ↪ g", dpi=220)
242     plt.close(fig)
243
244 # -----
245 ↪ -----
246 # Main
247 # -----
248 ↪ -----
249 def main():
250     parser = argparse.ArgumentParser(
251         description="Compare coherence descriptors vs
252 ↪ Morans I clustering strength"
253     )
254     parser.add_argument("--coherence", required=True)
255     parser.add_argument("--morans", required=True)
256     parser.add_argument("--output", required=True)
257
258     args = parser.parse_args()
259     os.makedirs(args.output, exist_ok=True)
260
261     df = load_data(args.coherence, args.morans)
262
263     print("\nDetected role distribution:\n",
264 ↪ df["ROLE"].value_counts(), "\n")
265
266     plot_global(df, args.output)
267     plot_combined_mean(df, args.output)
268
269     for _, sub in df.groupby("REGION"):
270         plot_region(sub, args.output)
271
272     print(f"\nPlots written to
273 ↪ {os.path.abspath(args.output)}")
274
275 if __name__ == "__main__":
276     main()

```

sks_{shear_m}isfit.py

```

1  #!/usr/bin/env python3
2
3  import numpy as np
4  import pandas as pd
5  from scipy.interpolate import RegularGridInterpolator
6  from tqdm import tqdm
7  import importlib.util
8
9  # -----
10 ↪ -----
11 # 1. LOAD SKS DATA (Silver 1996)
12 # -----
13 ↪ -----
14
15 sks = pd.read_csv("silver1996.csv", sep=";",
16 ↪ encoding="utf-8-sig")
17 sks.columns = sks.columns.str.strip()
18
19 lat_obs = sks["Latitude"].values
20 lon_obs = sks["Longitude"].values
21 phi_obs = sks["phi_deg"].values % 180.0
22
23 # -----
24 ↪ -----
25 # 2. IMPORT SHEAR FIELD FROM shear-map.py (UNCHANGED)
26 # -----
27 ↪ -----
28
29 spec = importlib.util.spec_from_file_location("shear_
30 ↪ map", "shear-map.py")
31 shear_map = importlib.util.module_from_spec(spec)
32 spec.loader.exec_module(shear_map)
33
34 lon_grid = shear_map.lon           # 1D, degrees
35 lat_grid = shear_map.lat           # 1D, degrees
36 theta_grid = shear_map.theta       # 2D, radians
37
38 # -----
39 ↪ -----
40 # 3. CONVERT TO GEOGRAPHIC AZIMUTH + CONJUGATE SHEAR
41 ↪ NETS

```

```

36 # -----
37 ↪ -----
38 # Convert mathematical angle (CCW from East) to
39 # geographic azimuth (CW from North), axial
40 theta0 = (90.0 - np.rad2deg(theta_grid)) % 180.0
41
42 theta_net1 = (theta0 + 45.0) % 180.0
43 theta_net2 = (theta0 - 45.0) % 180.0
44
45
46 interp_net1 = RegularGridInterpolator(
47     (lat_grid, lon_grid),
48     theta_net1,
49     bounds_error=False,
50     fill_value=np.nan
51 )
52
53 interp_net2 = RegularGridInterpolator(
54     (lat_grid, lon_grid),
55     theta_net2,
56     bounds_error=False,
57     fill_value=np.nan
58 )
59
60 def shear_net_azimuths(lon, lat):
61     return (
62         interp_net1((lat, lon)),
63         interp_net2((lat, lon))
64     )
65
66 # -----
67 ↪ -----
68 # 4. SAMPLE SHEAR NETS AT SKS SITES + MASK
69 # -----
70 ↪ -----
71
72 net1_at_sks = np.zeros(len(lon_obs))
73 net2_at_sks = np.zeros(len(lon_obs))
74
75 for i in range(len(lon_obs)):
76     net1_at_sks[i], net2_at_sks[i] =
77     ↪ shear_net_azimuths(
78         lon_obs[i], lat_obs[i]
79     )
80
81 valid = np.isfinite(net1_at_sks) &
82 ↪ np.isfinite(net2_at_sks)
83
84 print(f"Total SKS records: {len(valid)}")
85 print(f"Valid shear-field overlaps: {valid.sum()}")
86 print(f"Excluded (outside shear grid):
87 ↪ {(~valid).sum()}")
88
89 lon_obs = lon_obs[valid]
90 lat_obs = lat_obs[valid]
91 phi_obs = phi_obs[valid]
92 net1_at_sks = net1_at_sks[valid]
93 net2_at_sks = net2_at_sks[valid]
94
95 # -----
96 ↪ -----
97
98 # 5. AXIAL MISFIT FUNCTION
99 # -----
100 ↪ -----
101
102 def axial_misfit(theta_obs, theta_model):
103     d = np.abs(theta_obs - theta_model) % 180.0
104     return np.minimum(d, 180.0 - d)
105
106 # -----
107 ↪ -----
108
109 # 6. OBSERVED MISFIT (MINIMUM OF NET-1 / NET-2)
110 # -----
111 ↪ -----
112
113 misfit1 = axial_misfit(phi_obs, net1_at_sks)
114 misfit2 = axial_misfit(phi_obs, net2_at_sks)
115
116 obs_misfit = np.minimum(misfit1, misfit2)
117
118 obs_mean = obs_misfit.mean()
119 obs_median = np.median(obs_misfit)
120 obs_variance = 1.0 - np.mean(np.cos(np.deg2rad(2.0 *
121 ↪ obs_misfit)))
122
123 print("Observed mean misfit:", obs_mean)
124 print("Observed median misfit:", obs_median)
125 print("Observed axial variance:", obs_variance)
126
127 # -----
128 ↪ -----
129
130 # 7. NULL HYPOTHESIS: GLOBAL AXIAL ROTATION
131 # -----
132 ↪ -----
133
134 K = 10000
135 null_mean = np.zeros(K)
136 null_variance = np.zeros(K)
137
138 for k in range(K):
139     alpha = np.random.uniform(0.0, 180.0)
140     rotated_phi = (phi_obs + alpha) % 180.0
141
142     m1 = axial_misfit(rotated_phi, net1_at_sks)
143     m2 = axial_misfit(rotated_phi, net2_at_sks)
144     m = np.minimum(m1, m2)
145
146     null_mean[k] = m.mean()
147     null_variance[k] = 1.0 -
148     ↪ np.mean(np.cos(np.deg2rad(2.0 * m)))
149
150 # -----
151 ↪ -----
152
153 # 8. EMPIRICAL P-VALUES
154 # -----
155 ↪ -----
156
157 p_mean = np.mean(null_mean <= obs_mean)
158 p_var = np.mean(null_variance <= obs_variance)
159
160 print("p-value (mean misfit):", p_mean)
161 print("p-value (axial variance):", p_var)

```



```

148
149
150 # -----
151 ↪ -----
152 # 9. SAVE RESULTS
153 # -----
154 ↪ -----
155
156 out = sks.loc[valid].copy()
157 out["shear_net1_azimuth"] = net1_at_sks
158 out["shear_net2_azimuth"] = net2_at_sks
159 out["angular_misfit_deg"] = obs_misfit
160
161 out.to_csv("Silver1996_shear_misfit_results.csv",
162 ↪ index=False)
163 print("Saved: Silver1996_shear_misfit_results.csv")

```

silverplot.py

```

1  #!/usr/bin/env python3
2
3  import numpy as np
4  import pandas as pd
5  import matplotlib.pyplot as plt
6
7  # -----
8  ↪ -----
9  # 1. LOAD RESULTS FILE
10 # -----
11 ↪ -----
12
13 df =
14 ↪ pd.read_csv("Silver1996_shear_misfit_results.csv")
15
16 phi_obs = df["phi_deg"].values % 180.0
17 net1_at_sks = df["shear_net1_azimuth"].values
18 obs_misfit = df["angular_misfit_deg"].values
19
20 # -----
21 ↪ -----
22 # 2. AXIAL MISFIT FUNCTION
23 # -----
24 ↪ -----
25
26 def axial_misfit(theta_obs, theta_model):
27     d = np.abs(theta_obs - theta_model) % 180.0
28     return np.minimum(d, 180.0 - d)
29
30 # -----
31 ↪ -----
32 # 3. BUILD NULL HISTOGRAM ENVELOPE (SINGLE-NET)
33 # -----
34 ↪ -----
35
36 K = 10000
37 bins = np.linspace(0, 90, 31)
38 bin_centers = 0.5 * (bins[:-1] + bins[1:])
39
40 null_hist = np.zeros((K, len(bin_centers)))
41
42 for k in range(K):
43     alpha = np.random.uniform(0.0, 180.0)

```

```

39     rotated_phi = (phi_obs + alpha) % 180.0
40     misfit_k = axial_misfit(rotated_phi, net1_at_sks)
41     h, _ = np.histogram(misfit_k, bins=bins,
42 ↪ density=True)
43     null_hist[k] = h
44
45 null_lo = np.percentile(null_hist, 5, axis=0)
46 null_hi = np.percentile(null_hist, 95, axis=0)
47 null_med = np.percentile(null_hist, 50, axis=0)
48
49 # -----
50 ↪ -----
51 # 4. OBSERVED HISTOGRAM
52 # -----
53 ↪ -----
54
55 obs_hist, _ = np.histogram(obs_misfit, bins=bins,
56 ↪ density=True)
57
58 # -----
59 ↪ -----
60
61 plt.figure(figsize=(7, 4.5))
62
63 plt.fill_between(
64     bin_centers,
65     null_lo,
66     null_hi,
67     color="lightgray",
68     alpha=0.85,
69     label="Null envelope (5{95%}")
70 )
71
72 plt.plot(
73     bin_centers,
74     null_med,
75     color="gray",
76     linestyle="--",
77     linewidth=1.5,
78     label="Null median"
79 )
80
81 plt.step(
82     bin_centers,
83     obs_hist,
84     where="mid",
85     color="tab:orange",
86     linewidth=2.2,
87     label="Observed (Silver 1996)"
88 )
89
90 plt.axvline(
91     obs_misfit.mean(),
92     color="tab:orange",
93     linestyle=":",
94     linewidth=2,
95     label=f"Observed mean = {obs_misfit.mean():.1f}°"
96 )

```

```

97 plt.axvline(
98     45,
99     color="black",
100    linestyle=":",
101    linewidth=1,
102    label="Random axial mean (45°)"
103 )
104
105 plt.xlabel("Axial misfit angle (degrees)")
106 plt.ylabel("Probability density")
107 plt.xlim(0, 90)
108 plt.title("SKS Fast-Axis Misfit vs Shear-Net
109 ↪ Geometry")
110
111 plt.legend(frameon=False)
112 plt.tight_layout()
113 plt.show()

```

silverplot2.py <MINTED>

seisglob_full.py

```

1  #!/usr/bin/env python3
2  import numpy as np
3  import xarray as xr
4  from scipy.stats import ks_2samp
5  from tqdm import trange
6
7  # =====
8  ↪ =====
9  # CONFIGURATION
10 # =====
11 ↪ =====
12
13 NETCDF_FILE = "SEISGLOB2_percent.nc"
14
15 DEPTH_BANDS = {
16     "upper_mantle": (300, 600),
17     "mid_mantle": (900, 1200),
18     "lower_mantle": (1800, 2400),
19 }
20
21 PERCENTILES = [90, 95, 97.5]
22 N_NULL = 1000
23
24 # Euler points (degrees)
25 # (ON, 59W) -> 301E
26 # (ON, 121E)
27 EULERS = [
28     (0.0, 301.0),
29     (0.0, 121.0),
30 ]
31
32 # =====
33 ↪ =====
34 # GEOMETRY
35 # =====
36 ↪ =====
37
38 def great_circle_distance(lat1, lon1, lat2, lon2):
39     """
40     Angular great-circle distance in degrees.
41     Inputs in degrees.
42     """

```

```

39 lat1, lon1, lat2, lon2 = map(np.deg2rad, [lat1,
40 ↪ lon1, lat2, lon2])
41
42 cos_d = (
43     np.sin(lat1) * np.sin(lat2)
44     + np.cos(lat1) * np.cos(lat2) * np.cos(lon1 -
45 ↪ lon2)
46 )
47
48 cos_d = np.clip(cos_d, -1.0, 1.0)
49 return np.rad2deg(np.arccos(cos_d))
50
51 # =====
52 ↪ =====
53 # DATA EXTRACTION
54 # =====
55 ↪ =====
56
57 def extract_extremes(lat, lon, depth, dvs,
58 ↪ depth_range, percentile):
59     """
60     Extract extreme |dvs| points within a depth band.
61     Returns array of (lat, lon, depth, dvs).
62     """
63
64     dmin, dmax = depth_range
65     depth_mask = (depth >= dmin) & (depth <= dmax)
66
67     dvs_slice = dvs[depth_mask, :, :]
68     depth_vals = depth[depth_mask]
69
70     abs_vals =
71     ↪ np.abs(dvs_slice[np.isfinite(dvs_slice)])
72     threshold = np.percentile(abs_vals, percentile)
73
74     points = []
75
76     for k, z in enumerate(depth_vals):
77         layer = dvs_slice[k]
78         mask = np.abs(layer) >= threshold
79         lat_idx, lon_idx = np.where(mask)
80
81         for i, j in zip(lat_idx, lon_idx):
82             points.append((lat[i], lon[j], z, layer[i,
83 ↪ j]))
84
85     return np.array(points)
86
87 def compute_euler_distances(points):
88     """
89     Compute minimum angular distance to either Euler
90     ↪ point.
91     """
92
93     dists = []
94
95     for lat_p, lon_p, _, _ in points:
96         d = min(
97             great_circle_distance(lat_p, lon_p, e_lat,
98 ↪ e_lon)
99             for e_lat, e_lon in EULERS
100         )
101         dists.append(d)
102     return np.array(dists)

```

```

93 # ===== 144
94 ↪ ===== 145
95 # NULL MODELS 146
96 ↪ ===== 147
97 def longitude_randomised_null(points, n_iter): 148
98     """ 149
99     Longitude-randomised null (latitude preserved). 150
100     """ 151
101     null_dists = [] 152
102 153
103     for _ in trange(n_iter, desc="Longitude null"): 154
104         rand_lon = np.random.uniform(0, 360, 155
105             ↪ size=len(points)) 156
106         for (lat_p, _, _, _), lon_r in zip(points, 157
107             ↪ rand_lon): 158
108             d = min( 159
109                 great_circle_distance(lat_p, lon_r, 160
110                 ↪ e_lat, e_lon) 161
111                 for e_lat, e_lon in EULERS 162
112             ) 163
113             null_dists.append(d) 164
114 165
115     return np.array(null_dists) 166
116 167
117 def lat_symmetrised_null(points, n_iter): 168
118     """ 169
119     Latitude-symmetrised + longitude-randomised null. 170
120     Preserves |latitude|. 171
121     """ 172
122     null_dists = [] 173
123 174
125     for _ in trange(n_iter, desc="Lat-sym null"): 175
126         rand_lon = np.random.uniform(0, 360, 176
127             ↪ size=len(points)) 177
128         rand_sign = np.random.choice([-1, 1], 178
129             ↪ size=len(points)) 179
130 180
131         for (lat_p, _, _, _), lon_r, sgn in 181
132             ↪ zip(points, rand_lon, rand_sign): 182
133             lat_r = sgn * abs(lat_p) 183
134             d = min( 184
135                 great_circle_distance(lat_r, lon_r, 185
136                 ↪ e_lat, e_lon) 186
137                 for e_lat, e_lon in EULERS 187
138             ) 188
139             null_dists.append(d) 189
140 190
141     return np.array(null_dists) 191
142 192
143 # ===== 193
144 ↪ ===== 194
145 # ANALYSIS DRIVER 195
146 # ===== 196
147 ↪ ===== 197
148 198
149 def run_test(lat, lon, depth, dvs, band_name, 199
150     ↪ depth_range, percentile): 200
151     print(f"\n=== {band_name.upper()} | {percentile}th 201
152     ↪ percentile ===") 202
153 203
154     points = extract_extremes(lat, lon, depth, dvs, 204
155     ↪ depth_range, percentile) 205
156     obs_dists = compute_euler_distances(points) 206
157 207
158     print(f"N anomalies: {len(points)}") 208
159     print(f"Observed mean dist: 209
160     ↪ {np.mean(obs_dists):.2f}") 210
161 211
162     # Longitude null 212
163     null_lon = longitude_randomised_null(points, 213
164     ↪ N_NULL) 214
165     ks_lon = ks_2samp(obs_dists, null_lon) 215
166 216
167     print("\n[Longitude-randomised null]") 217
168     print(f"Null mean dist: {np.mean(null_lon):.2f}°") 218
169     print(f"mean (obs-null): {np.mean(obs_dists) - 219
170     ↪ np.mean(null_lon):.2f}") 220
171     print(f"KS: {ks_lon.statistic:.3f} | p = 221
172     ↪ {ks_lon.pvalue:.3e}") 222
173 223
174     # Latitude-symmetrised null 224
175     null_lat = lat_symmetrised_null(points, N_NULL) 225
176     ks_lat = ks_2samp(obs_dists, null_lat) 226
177 227
178     print("\n[Latitude-symmetrised null]") 228
179     print(f"Null mean dist: {np.mean(null_lat):.2f}°") 229
180     print(f"mean (obs-null): {np.mean(obs_dists) - 230
181     ↪ np.mean(null_lat):.2f}") 231
182     print(f"KS: {ks_lat.statistic:.3f} | p = 232
183     ↪ {ks_lat.pvalue:.3e}") 233
184 234
185 # ===== 235
186 ↪ ===== 236
187 # MAIN 237
188 # ===== 238
189 ↪ ===== 239
190 240
191 def main(): 241
192     ds = xr.open_dataset(NETCDF_FILE) 242
193 243
194     lat = ds["latitude"].values 244
195     lon = ds["longitude"].values 245
196     depth = ds["depth"].values 246
197     dvs = ds["dvs"].values 247
198 248
199     for band, depth_range in DEPTH_BANDS.items(): 249
200         for p in PERCENTILES: 250
201             run_test(lat, lon, depth, dvs, band, 251
202             ↪ depth_range, p) 252
203 253
204     if __name__ == "__main__": 254
205         main() 255
206 256
207 seisglob2midmantlecdfcomposite.py 257
208 258
209 1 #!/usr/bin/env python3 259
210 2 import numpy as np 260
211 3 import xarray as xr 261
212 4 import matplotlib.pyplot as plt 262
213 5 from tqdm import trange 263
214 6 264

```

```

7  # =====
8  # CONFIGURATION
9  # =====
10
11  NETCDF_FILE = "SEISGLOB2_percent.nc"
12  MID_MANTLE_RANGE = (900, 1200)
13  PERCENTILES = [90, 95, 97.5]
14  N_NULL = 1000
15
16  EULERS = [
17      (0.0, 301.0), # ON, 59W
18      (0.0, 121.0), # ON, 121E
19  ]
20
21  # =====
22  # GEOMETRY
23  # =====
24
25  def great_circle_distance(lat1, lon1, lat2, lon2):
26      lat1, lon1, lat2, lon2 = map(np.deg2rad, [lat1,
27          ↪ lon1, lat2, lon2])
28      cos_d = (
29          np.sin(lat1) * np.sin(lat2)
30          + np.cos(lat1) * np.cos(lat2) * np.cos(lon1 -
31          ↪ lon2)
32      )
33      cos_d = np.clip(cos_d, -1.0, 1.0)
34      return np.rad2deg(np.arccos(cos_d))
35  # =====
36  # DATA EXTRACTION
37  # =====
38
39  def extract_extremes(lat, lon, depth, dvs,
40      ↪ depth_range, percentile):
41      dmin, dmax = depth_range
42      depth_mask = (depth >= dmin) & (depth <= dmax)
43
44      dvs_slice = dvs[depth_mask, :, :]
45      depth_vals = depth[depth_mask]
46
47      abs_vals =
48      ↪ np.abs(dvs_slice[np.isfinite(dvs_slice)])
49      threshold = np.percentile(abs_vals, percentile)
50
51      points = []
52
53      for k, z in enumerate(depth_vals):
54          layer = dvs_slice[k]
55          mask = np.abs(layer) >= threshold
56          lat_idx, lon_idx = np.where(mask)
57
58          for i, j in zip(lat_idx, lon_idx):
59              points.append((lat[i], lon[j]))
60
61      return np.array(points)
62
63  def compute_euler_distances(points):
64      dists = []
65      for lat_p, lon_p in points:
66          d = min(
67              great_circle_distance(lat_p, lon_p, e_lat,
68              ↪ e_lon)
69              for e_lat, e_lon in EULERS
70          )
71          dists.append(d)
72      return np.array(dists)
73  # =====
74  # NULL MODELS
75  # =====
76
77  def longitude_randomised_null(points, n_iter):
78      null_dists = []
79      for _ in trange(n_iter, desc="Longitude null"):
80          rand_lon = np.random.uniform(0, 360,
81          ↪ size=len(points))
82          for (lat_p, _), lon_r in zip(points,
83          ↪ rand_lon):
84              d = min(
85                  great_circle_distance(lat_p, lon_r,
86                  ↪ e_lat, e_lon)
87                  for e_lat, e_lon in EULERS
88              )
89              null_dists.append(d)
90      return np.array(null_dists)
91
92  def lat_symmetrised_null(points, n_iter):
93      null_dists = []
94      for _ in trange(n_iter, desc="Lat-sym null"):
95          rand_lon = np.random.uniform(0, 360,
96          ↪ size=len(points))
97          rand_sign = np.random.choice([-1, 1],
98          ↪ size=len(points))
99
100          for (lat_p, _), lon_r, sgn in zip(points,
101          ↪ rand_lon, rand_sign):
102              lat_r = sgn * abs(lat_p)
103              d = min(
104                  great_circle_distance(lat_r, lon_r,
105                  ↪ e_lat, e_lon)
106                  for e_lat, e_lon in EULERS
107              )
108              null_dists.append(d)
109      return np.array(null_dists)
110
111  # =====
112  # CDF UTILITIES
113  # =====
114
115  def empirical_cdf(data):
116      x = np.sort(data)
117      y = np.arange(1, len(x) + 1) / len(x)

```



```

113     return x, y
114
115
116 # =====
117 ↪ =====
118 # MAIN FIGURE
119 ↪ =====
120
121 def main():
122     ds = xr.open_dataset(NETCDF_FILE)
123     lat = ds["latitude"].values
124     lon = ds["longitude"].values
125     depth = ds["depth"].values
126     dvs = ds["dvs"].values
127
128     fig, axes = plt.subplots(
129         1, 3, figsize=(14, 4), sharex=True,
130         ↪ sharey=True
131     )
132
133     for ax, p, label in zip(
134         axes, PERCENTILES, ["(a)", "(b)", "(c)"]
135     ):
136         print(f"\nProcessing {p}th percentile")
137
138         points = extract_extremes(
139             lat, lon, depth, dvs, MID_MANTLE_RANGE, p
140         )
141
142         obs = compute_euler_distances(points)
143         null_lon = longitude_randomised_null(points,
144             ↪ N_NULL)
145         null_lat = lat_symmetrised_null(points,
146             ↪ N_NULL)
147
148         x_obs, y_obs = empirical_cdf(obs)
149         x_lon, y_lon = empirical_cdf(null_lon)
150         x_lat, y_lat = empirical_cdf(null_lat)
151
152         ax.plot(x_obs, y_obs, linewidth=2,
153             ↪ label="Observed")
154         ax.plot(x_lon, y_lon, linestyle="--",
155             ↪ label="Longitude null")
156         ax.plot(x_lat, y_lat, linestyle=":",
157             ↪ label="Lat-sym null")
158
159         ax.set_title(f"{label} {p}th percentile")
160         ax.grid(alpha=0.3)
161
162     axes[0].set_ylabel("Cumulative probability")
163     for ax in axes:
164         ax.set_xlabel("Angular distance to nearest
165             ↪ Euler (degrees)")
166
167     # Single legend
168     handles, labels =
169     ↪ axes[0].get_legend_handles_labels()
170     fig.legend(
171         handles, labels,
172         loc="lower center",
173         ncol=3,
174         frameon=False
175     )
176
177     plt.tight_layout(rect=[0, 0.12, 1, 1])
178     plt.show()
179
180 if __name__ == "__main__":
181     main()

```