

# Implicit Global Frontiers: Benchmarking Neural Implicit Representations on Political Spherical Geometry

Freitas Armando  
University of Luxembourg  
Email: armando.freitas.001@student.uni.lu

This report has been produced under the supervision of:

Nicolas Navet  
University of Luxembourg  
Email: nicolas.navet@uni.lu

## Abstract

*Neural Implicit Representations (NIRs) have achieved state-of-the-art results in compressing visual and geometric data by parameterizing signals as continuous functions. However, their application to global geospatial data, characterized by spherical topology and extreme frequency variations (e.g., complex coastlines vs. open oceans), remains under-explored. This work investigates the efficacy of NIRs in approximating the global spherical distance-to-border field and political classification labels within a strict 8MB parameter budget. We benchmark twelve modern architectures, including SIREN, INCODE, HOSC, Random Fourier Features and MFN, using a hybrid geodesic normal displacement sampling strategy on the unit sphere. Contrary to the prevailing trend of using deep MLPs with periodic activations, our results demonstrate that **Multiplicative Filter Networks (MFN)** utilizing Gabor filters achieve the best balance between classification accuracy and regression stability. We identify a critical "Spectral Wall": while Random Fourier Features (RFF) successfully capture high-frequency border details (< 4km error), they suffer from catastrophic global incoherence (> 150km RMSE). Conversely, smooth architectures like HOSC achieve low global error but fail to resolve complex borders. We conclude that monolithic global networks are insufficient for planetary-scale GIS, motivating the need for spatially partitioned neural architectures. The code is available at <https://github.com/fre-ar/bsp-af-s5>*

## 1. Introduction

Digital representations of the Earth's political geography are fundamental to Geographic Information Systems (GIS), offline navigation, and spatial analysis. Traditionally, these boundaries are stored as explicit vector geometries (polygons and linestrings). While exact, vector representations scale linearly in memory complexity with geometric detail; high-resolution coastlines (e.g., Norway) require orders of magnitude more storage than smooth borders (e.g., Western Sahara). As applications demand higher fidelity on resource-constrained devices, the trade-off between resolution and storage becomes a critical bottleneck.

Neural Implicit Representations (NIRs, sometimes INRs), or Coordinate-Based Networks, propose a paradigm shift: representing geometry not as a discrete mesh or list of vertices, but as a continuous function  $f_\theta(\mathbf{x})$  parameterized by a neural network [1]. By learning the weights  $\theta$ , NIRs

can theoretically encode signals at infinite resolution with a fixed memory footprint, though this capability is practically bounded by the network's spectral bandwidth. This approach has revolutionized 3D reconstruction and novel view synthesis [2, 3].

However, applying NIRs to planetary-scale political geography presents unique challenges distinct from standard 3D object reconstruction. First, the domain is strictly spherical ( $\mathbb{S}^2$ ), requiring metrics and sampling strategies adapted to non-Euclidean topology. Second, the signal exhibits extreme spectral variance: vast regions of constant value (oceans, large country interiors) are punctuated by high-frequency discontinuities (fractal coastlines, enclaves). This creates a severe test for the known *spectral bias* of neural networks: the tendency of gradient descent to prioritize low-frequency components, resulting in over-smoothed approximations of sharp features [4].

In this work, we investigate whether current State-of-the-Art (SotA) NIR architectures can efficiently approximate a complex spherical distance field tied to political geography. We task the networks to jointly predict: (a) the geodesic distance to the nearest international border, (b) the containing country, and (c) the adjacent country. We impose a strict practical constraint: the model size must not exceed 8MB, targeting lightweight mobile deployment.

We benchmark twelve architectures, ranging from periodic activation networks (SIREN [3], FINER [5]) to filter-based networks (MFN [6]) and hybrid conditioning models (INCODE [7]). To address the sparsity of borders on the globe, we propose a *Geodesic Normal Displacement* sampling strategy that balances global coverage with high-frequency boundary refinement.

Our contributions are as follows:

- 1) A formulation of multi-task political classification and spherical distance fields as a neural implicit learning problem.
- 2) A rigorous benchmark of twelve NIR architectures on

- planetary-scale data, analyzing the trade-offs between spectral bandwidth and global coherence.
- 3) Empirical evidence that global, non-partitioned MLPs, even with advanced frequency-tuning mechanisms, are insufficient to capture the full frequency spectrum of Earth’s political borders within a compact budget, motivating future research into spherical spatial partitioning. item An open-source, reproducible PyTorch codebase for benchmarking spherical neural fields, available at <https://github.com/fre-ar/bsp-af-s5>.

## 2. Related Work

The application of neural networks to represent continuous geometric signals relies on overcoming the inherent limitations of standard Multi-Layer Perceptrons (MLPs) in capturing high-frequency details. This section reviews the progression of Neural Implicit Representations (NIRs) from simple coordinate networks to complex hybrid architectures, categorizing them by their strategy to mitigate spectral bias.

### 2.1. Spectral Bias and Input Encodings

A fundamental challenge in training coordinate-based networks is the *spectral bias* phenomenon, where standard MLPs equipped with ReLU activations tend to learn low-frequency components of functions significantly faster than high-frequency components [4]. In the context of geospatial data, this results in over-smoothed coastlines and merged enclaves. To counteract this, input coordinate mapping strategies were developed.

**Basic and Positional Encodings:** Early approaches mapped input coordinates  $\mathbf{x}$  to a higher dimensional space using fixed trigonometric functions  $\gamma(\mathbf{x}) = [\cos(2\pi\mathbf{x}), \sin(2\pi\mathbf{x})]$ . Mildenhall et al. [8] extended this with *Positional Encoding* (PE)  $\gamma(\mathbf{x}) = [\dots, \cos(2\pi\sigma^{j/m}\mathbf{x}), \sin(2\pi\sigma^{j/m}\mathbf{x}), \dots]$  for  $j \in \{0, \dots, m-1\}$ . Using a deterministic frequency bank, it allows the network to interpolate data at multiple scales.

**Fourier Features:** Tancik et al. [2] generalized this by projecting inputs via a random  $m \times d$  matrix  $\mathbf{B}$  drawn from a Gaussian distribution  $\mathcal{N}(0, \sigma^2)$ :  $\gamma(\mathbf{x}) = [\cos(2\pi\mathbf{B}\mathbf{x}), \sin(2\pi\mathbf{B}\mathbf{x})]$ . This effectively transforms the Neural Tangent Kernel (NTK) into a stationary kernel with a tunable bandwidth  $\sigma$ , offering better control over the learnable frequency spectrum than deterministic PE. While effective for image regression, static Fourier mappings require careful tuning of the bandwidth parameter  $\sigma$ ; a  $\sigma$  too low results in blurring, while a  $\sigma$  too high introduces high-frequency noise artifacts.

### 2.2. Periodic and Advanced Activations

An alternative to input encoding is modifying the activation function itself.

**Periodic Activations:** Sitzmann et al. [3] introduced **SIREN**, replacing ReLU with  $\phi(x) = \sin(\omega x)$ . The derivatives of a SIREN network remain well-behaved sine functions,

allowing it to model complex gradients and differential equations naturally. However, SIREN is highly sensitive to initialization and the choice of the global frequency hyperparameter  $\omega$ .

To address this, recent works have sought to improve the stability and spectral flexibility of periodic activations. **FINER** [5] introduces a variable-periodic activation function  $\phi(x) = \sin(\omega(|x| + 1)x)$ . By initializing bias terms appropriately, it allows different neurons to operate in different frequency bands of the sine function. This effectively allows the network to tune its spectral bias dynamically.

Similarly, **HOSC** [9] employs a hyperbolic oscillation function  $\phi(x) = \tanh(\beta \sin(\omega x))$ , introducing a sharpness parameter  $\beta$  to better capture sharp features (sudden signal changes) while maintaining smooth transitions in low-frequency regions.

From a signal processing perspective, Saratchandran et al. [10] argue that the **Sinc** function  $\phi(x) = \text{sinc}(\omega x) = \frac{\sin(\omega x)}{\omega x}$  is theoretically optimal for band-limited signal reconstruction, though its empirical performance on complex non-Euclidean manifolds remains less explored.

**Windowed and Complex Activations:** To improve training stability, Ramasinghe & Lucey [11] proposed **GAUSS**, using a non-periodic activation function  $\phi(x) = e^{-(sx)^2}$ , which acts as a local support bump function with a gaussian window width hyperparameter  $s$ .

Saragadam et al. [12] introduced **WIRE**, utilizing a complex Gabor wavelet  $\phi(x) = e^{i\omega x - (sx)^2}$ , combining the frequency-capturing properties of SIREN with the spatial locality of GAUSS.

### 2.3. Architectural Innovations and Training

Beyond single-neuron modifications, recent works propose macro-architectural changes.

**Hybrid Conditioning:** **INCODE** [7] separates frequency control from coordinate processing. It employs a dual-network structure: a "Harmonizer" network takes a learnable latent code  $z$  (not the coordinates) to predict the optimal modulation parameters (frequencies and phase shifts), which are then injected into the "Composer" network's activation functions. This allows for dynamic, task-specific spectral tuning.

**Filter Networks: Multiplicative Filter Networks (MFN)** [6] abandon standard depth composition. Instead, they apply a recursive Hadamard product of sinusoidal or Gabor filters at each layer, behaving akin to a Fourier series expansion, accumulating frequency components rather than composing them.

**Fourier Reparameterization (FR):** Addressing the optimization landscape directly, Shi et al. [13] propose FR Training. Rather than changing the architecture, FR decomposes the weight matrices  $\mathbf{W}$  into a fixed Fourier basis  $\mathbf{B}$  and learnable coefficients  $\mathbf{A}$  ( $\mathbf{W} = \mathbf{AB}$ ). This reparameterization aligns the gradient flow with the frequency spectrum of the target signal, accelerating convergence for high-frequency details.

## 2.4. Geospatial Gap Analysis

Despite these advances, existing benchmarks are primarily conducted on distinct, single-scale objects (e.g., single-room SDFs) or 2D images. Global political geography presents a fundamentally different challenge: it is defined on a sphere ( $\mathbb{S}^2$ ) and exhibits extreme multi-scale variation, from the smooth, continental borders of Africa to the fractal, island-heavy coastlines of Scandinavia. Furthermore, most NIR approaches optimize for visual fidelity (PSNR) or standard SDF reconstruction. They rarely tackle the joint multi-task problem of exact distance regression coupled with strict classification (country membership) under a tight mobile-grade memory budget (< 8MB). This work evaluates how these diverse SotA architectural strategies, from Gabor wavelets (WIRE) to dynamic conditioning (INCODE), perform under the strict constraints of high-precision geospatial classification and distance regression.

## 3. Methodology

We formulate the task of learning global political geography as a multi-task function approximation problem on the unit sphere. This section defines the mathematical problem, the dataset generation pipeline, the hybrid sampling strategy, and the unified neural architecture used for benchmarking.

### 3.1. Problem Formulation

Let  $\mathcal{C} = \{c_1, \dots, c_N, c_{water}\}$  be the set of discrete political entities (countries), including a special label  $c_{water}$  representing all non-territorial water bodies to ensure a complete partition of the globe. Let  $\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x}\|_2 = 1\}$  denote the unit sphere.

We seek to learn a function  $f_\theta : \mathbb{S}^2 \rightarrow \mathbb{R}_{\geq 0} \times \mathcal{C} \times \mathcal{C}$  parameterized by weights  $\theta$ . For any query point  $\mathbf{x} \in \mathbb{S}^2$ , the network predicts a tuple  $(\hat{d}, \hat{c}_{in}, \hat{c}_{adj})$ :

- $\hat{d}$ : The geodesic distance to the nearest international border segment.
- $\hat{c}_{in}$ : The classification label of the country containing  $\mathbf{x}$ .
- $\hat{c}_{adj}$ : The classification label of the neighbor country sharing the nearest border segment.

### 3.2. Data Representation and Ground Truth

Our source data consists of the ADM0 (national) boundaries from the World Bank dataset. To generate ground truth values for training, we implement a high-performance geometric query engine. The boundaries are pre-processed into a collection of geodesic arcs. We construct a spherical K-Dimensional Tree (KDTree) of arc midpoints to accelerate nearest-neighbor queries. For an arbitrary sample  $\mathbf{x}$ , the exact geodesic distance  $d(\mathbf{x})$  is computed by projecting  $\mathbf{x}$  onto the closest arc segment. Unlike standard 2D projections (e.g., Mercator), all computations are performed in 3D Cartesian coordinates to preserve spherical exactness.

Further details regarding dataset construction are included in Appendix B 9.

### 3.3. Geodesic Sampling Strategy

A naive uniform sampling of the sphere would result in a dataset dominated by the low-frequency geometry of open oceans and large continental interiors. To capture the high-frequency geometry of complex borders, we employ a hybrid sampling strategy  $D = D_{uni} \cup D_{near}$  with a 50/50 split:

**Global Uniform Sampling** ( $D_{uni}$ ): 50% of points are sampled uniformly from  $\mathbb{S}^2$ . These samples ensure the model learns the global topology and correct classification in deep interiors far from boundaries.

**Geodesic Normal Displacement** ( $D_{near}$ ): The remaining 50% target the high-frequency boundary regions. We sample a point  $\mathbf{b}$  on a random border segment and a displacement distance  $\delta$ . The training sample  $\mathbf{x}$  is generated by moving  $\delta$  km away from  $\mathbf{b}$  along the geodesic normal vector. To ensure scale invariance, supervising microscopic features (meters) as heavily as macroscopic features (kilometers),  $\delta$  is sampled from a log-uniform distribution:

$$p(\delta) \propto \frac{1}{\delta}, \quad \delta \in [0.1\text{km}, 25\text{km}] \quad (1)$$

This strategy forces the network to resolve geometry at multiple scales of magnitude, which is critical for resolving complex coastal fractals and small enclaves.

Empirically, this means that 22.62% of sampled points lie on the borders, and 53.67% are within 25km from borders, as opposed to 0.48% and 7.35% respectively if all the points were globally uniformly sampled.

### 3.4. Unified Network Architecture

To isolate the effect of the architecture (activation functions and encodings) from model capacity, we enforce a standardized topology for all benchmarks. All networks share a common **Trunk** consisting of a Multi-Layer Perceptron (MLP) with the specific architecture's activation function (e.g., Sine, Gaussian). The trunk maps the input coordinate  $\mathbf{x} \in \mathbb{R}^3$  to a latent feature vector  $\mathbf{z} \in \mathbb{R}^h$ . This latent vector is fed into three separate lightweight linear **Heads**:

- 1) **Regression Head**: Outputs a scalar  $\hat{y}_{dist}$ .
- 2) **Classification Head 1**: Outputs logits for  $c_{in}$  (size  $|\mathcal{C}|$ ).
- 3) **Classification Head 2**: Outputs logits for  $c_{adj}$  (size  $|\mathcal{C}|$ ).

Unless otherwise specified, we fix the budget to approximately 2 million parameters to simulate a resource-constrained mobile environment.

### 3.5. Loss Function

We train the networks using a composite loss function combining regression and classification objectives.

**Log-Distance Regression**: The dynamic range of geodesic distances is extreme (0 to  $\approx 5000$  km). Direct regression of

kilometers leads to unstable gradients. We instead regress the log-transformed distance:

$$y_{dist} = \ln(d(\mathbf{x}) + 1) \quad (2)$$

The regression loss  $\mathcal{L}_{reg}$  is the Mean Squared Error (MSE) between the predicted and ground truth log-distances. This transformation compresses the output space and naturally penalizes relative errors more heavily near zero (the borders).

**Total Loss:** The final objective is the unweighted sum of the regression loss and the Cross-Entropy (CE) losses for both classification tasks:

$$\mathcal{L}_{total} = \mathcal{L}_{reg} + \mathcal{L}_{CE}(c_{in}) + \mathcal{L}_{CE}(c_{adj}) \quad (3)$$

Empirically, we found that the magnitude of the log-distance error naturally balances with the cross-entropy terms, removing the need for hyperparameter tuning of loss weights.

## 4. Experiments

We evaluate twelve distinct Neural Implicit Representation architectures on the task of spherical political geography. This section details the benchmarked models, the training protocol, and the evaluation metrics used to quantify performance.

### 4.1. Benchmark Architectures

We selected a diverse set of architectures representing the current state-of-the-art in coordinate-based deep learning. All models are configured to strictly adhere to the  $\approx 8\text{MB}$  parameter budget (approx. 2M parameters in 32-bit float), translating to 8 layers (6 hidden) of 512 neurons each. For architectural hyperparameters (e.g., frequency multipliers  $\omega$ , bandwidths  $\sigma$ ), we adopt the recommended values reported in the respective original papers. Where authors provide different settings for different domains, we prioritize configurations tuned for 3D reconstruction tasks; if unavailable, we default to 2D image reconstruction settings.

- 1) **ReLU MLP (Baseline):** A standard Multi-Layer Perceptron with Rectified Linear Unit activations. We evaluate this with three encoding strategies: No Encoding, Positional Encoding (PE) [8], and Gaussian Random Fourier Features (RFF) [2]. For PE,  $\sigma = 5$  and  $m = 256$ . For RFF,  $\sigma = 10$  and  $m = 256$ .
- 2) **SIREN [3]:** A fully periodic MLP using  $\sin(\omega\mathbf{x})$ . We use the recommended  $\omega = 30$  and the authors' proposed initialization scheme to maintain activation variance.
- 3) **FINER [5]:** Uses variable-periodic activations  $\sin(\omega(|x| + 1)x)$  to increase spectral flexibility. The bias terms are initialized uniformly within range  $[-k, k]$  (with  $k = 20$ ) to force neurons into different frequency bands. We use the recommended  $\omega = 1$ .
- 4) **HOSC [9]:** Uses the hyperbolic oscillation activation  $\tanh(\beta \sin(\omega x))$  to capture sharp discontinuities. We use a proposed variant called AdaHOSC, where  $\beta$  is a learnable parameter for each layer. As recommended,

we initialize  $\beta = 8$  and use  $\omega = 30$  for the first layer, and  $\omega = 1$  for all others. Weights use a SIREN-style initialization.

- 5) **WIRE [12]:** Uses complex-valued Gabor wavelets  $e^{i\omega x - (sx)^2}$  to combine frequency selection with spatial locality. We use the recommended  $s = 20$  and  $\omega = 15$ .
- 6) **GAUSS [11]:** Uses Gaussian activations  $e^{-(sx)^2}$  for robust, initialization-invariant training. We use the recommended  $s = \frac{1}{0.1 \times \sqrt{2}} \approx 7.07$ .
- 7) **SINC [10]:** Uses the theoretically optimal band-limited reconstruction kernel  $\text{sinc}(\omega x) = \frac{\sin(\omega x)}{\omega x}$ . We use the recommended  $\omega = \pi$ .
- 8) **MFN (Fourier & Gabor) [6]:** Two variants of Multiplicative Filter Networks using sinusoidal and Gabor filters, respectively. While all other MLPs feature 8 layers of 512 neurons, to keep within the parameter budget, the MFN configurations feature only 7 layers.
- 9) **INCODE [7]:** A hybrid architecture where a "Harmonizer" network predicts dynamic activation parameters for the "Composer" network based on a learned latent code. We include the authors' auxiliary regularization loss  $\mathcal{L}_{reg} = \sum_i \lambda_i (\text{param}_i - \text{identity})^2$  to stabilize the modulation parameters. We use  $\omega = 30$
- 10) **Fourier Reparameterization (FR) [13]:** A training-time technique where weight matrices are decomposed as  $\mathbf{W} = \Lambda \mathbf{B}$ , where  $\mathbf{B}$  is a fixed Fourier basis and  $\Lambda$  are learnable coefficients. While the training-time parameter count is higher (12.8M), the inference-time model collapses  $\Lambda \mathbf{B}$  into a single matrix, adhering to the standard 1.88M budget. We apply this to a standard SIREN-style Sine-activation network, with the author's recommended  $\omega = 5$ .

Further details regarding specific initializations and hyperparameter choices are included in Appendix A 8.

### 4.2. Implementation Details and Optimization Sweep:

**Training Protocol:** All models are implemented in PyTorch.<sup>1</sup> We optimize using **Adam** with no weight decay. The training duration is fixed at 48,840 steps with a batch size of 4,096 points, equivalent to 200 million sample observations (seen exactly once). This "streaming" regime prevents overfitting to specific cached points and forces the network to learn the underlying function  $f_\theta$  rather than memorizing a finite set.

To ensure a fair comparison and mitigate optimization bias, we conduct a learning rate sweep for every architecture. We train four independent instances of each model with learning rates  $\text{LR} \in \{1 \times 10^{-5}, 4 \times 10^{-5}, 1 \times 10^{-4}, 4 \times 10^{-4}\}$ . The results reported in Table 1 correspond to the specific run that minimized the Validation Score ( $\mathcal{S}_{val}$ ). This ensures that each architecture is evaluated near its peak optimization performance.

1. Code available at <https://github.com/fre-ar/bsp-af-s5>

**Hardware:** Experiments were conducted on a single NVIDIA RTX 4060 GPU (8GB VRAM).

### 4.3. Evaluation Metrics

We assess performance using three primary metrics, evaluated on a held-out test set of 10 million points sampled with a 100% globally uniform sampling strategy (akin to  $D_{uni}$ ).

**1. Distance Accuracy (Regression):** Since global distances vary from 0 to 5000 km, Mean Squared Error is sensitive to outliers. We prioritize the **Median Absolute Error (MedAE)** of the re-transformed linear distance (in km) as a robust measure of central tendency. To provide a complete picture of the regression performance, we also report RMSE and 95th Percentile Error. Additionally, we report the **Border-MedAE**, computed only on the subset of points within 25km of a border, to measure high-frequency precision.

**2. Classification Accuracy:** We report both the **Accuracy** and **Balanced Accuracy** for both the containing country ( $c_{in}$ ) and the adjacent country ( $c_{adj}$ ). Balanced accuracy is the arithmetic mean of sensitivity per class, ensuring that small nations (e.g., Vatican City, Monaco) contribute equally to the score as large nations (e.g., Russia).

**3. Validation Score:** To rank the architectures with a single scalar, we define a composite heuristic:

$$S_{val} = \text{RMSE}_{km} + 100 \cdot ((1 - \text{BalAcc}_{c_{in}}) + (1 - \text{BalAcc}_{c_{adj}})) \quad (4)$$

This score penalizes distance error linearly and classification error heavily (1% accuracy loss  $\approx 1$  km distance error), reflecting the requirement for high-reliability labeling.

## 5. Results

We evaluate the twelve architectures on the held-out test set of 10 million points. Table 1 summarizes the global performance metrics.

### 5.1. Quantitative Benchmark

**Overall Performance Trends:** Contradicting our initial hypothesis that dynamic activation functions (INCODE, FINER) would dominate, **Multiplicative Filter Networks (MFN-Gabor)** achieved the best overall Validation Score (86.29). It demonstrated a superior balance between global regression stability and classification accuracy. **MFN-Fourier** achieved the highest classification accuracy of all models (99.5% Country Accuracy, 99.2% Neighbor Accuracy), suggesting that Fourier filters excel at partitioning the sphere but struggle with the continuous gradient required for distance regression (101 km RMSE).

**HOSC** and **INCODE** followed closely, with HOSC achieving the lowest global RMSE (54.85 km), and INCODE achieving the best Median Absolute Error (20.57 km). **SIREN** proved highly competitive, achieving a remarkably

low Border-MedAE of 4.50 km, outperforming newer architectures like WIRE and GAUSS.

**The Stability of Variable Periodicity:** A striking finding is the catastrophic failure of **FINER**, which failed to converge (Accuracy  $\approx 0.1\%$ ). We hypothesize that the variable-frequency term  $\omega(|x| + 1)x$  introduced instability in the gradients when applied to the complex topology of the sphere. In contrast, **INCODE**, which separates frequency modulation into a distinct "Harmonizer" network, remained stable and achieved the lowest Median Absolute Error.

**The Precision-Coherence Trade-off (ReLU + RFF):** **ReLU+RFF** provides the most critical insight into the Spectral Bias. It was the *only* architecture to meet the strict high-frequency requirement, achieving a **Border-MedAE of 3.95 km**. However, this came at the cost of global structure: its Global RMSE exploded to 153.98 km (nearly 3× that of HOSC). This indicates that Random Fourier Features allowed the network to memorize high-frequency border details effectively but prevented it from interpolating the smooth distance field in open regions.

### 5.2. Qualitative Analysis

To better understand the numerical performance, we visually inspect the learned fields of the top-performing models.

**Classification Decision Boundaries:** Figure 1 visualizes the country predictions of **MFN-Fourier**, the most accurate classifier. The model reproduces political borders with remarkable sharpness, correctly identifying small enclaves and complex coastlines. Figure 2 visualizes the country prediction error of **MFN-Fourier** as a scatter plot of the 10 million evaluation points. We observe that classification errors are not randomly distributed but are concentrated almost exclusively at the decision boundaries (borders). This effectively manifests as "anti-aliasing" noise, where the model is uncertain within a narrow band of a few kilometers around the true border.

Figure 3 visualizes the country prediction error of **MFN-Fourier** around the Alpes region of Europe. The model is capable of recognizing small details, such as the presence of the country of Liechtenstein, but struggles at producing sharp borders at the local level, showing the "anti-aliasing" effect once more.

Figure 4 visualizes the neighbor prediction of **MFN-Fourier**, showing similar accuracy as country classification.

**Distance Field & Error Distribution:** Figure 5 shows the signed distance field predicted by the overall winner, **MFN-Gabor**. The model successfully learns a smooth gradient field that increases monotonically with distance from the borders. However, inspecting the error residuals illustrated in Figure 6 reveals a distinct spatial pattern: while the log-distance loss function successfully suppresses relative error near borders (keeping Border-MedAE low), the absolute error (in km) naturally grows as the distance increases. The highest absolute errors are observed in the middle of the Pacific Ocean and Antarctica. This is an expected consequence of the objective function: minimizing  $\ln(d + 1)$  implies that a 10km error at

TABLE 1: **Benchmarking Results (Best of 4 LR Sweep).** Models are sorted by their aggregate Validation Score (lower is better). LR denotes the learning rate used for the reported run ( $1e^{-4} = 10^{-4}$ ). *MedAE* is Median Absolute Error in km. Border *MedAE* is *MedAE* measured within 25 km of international borders. *P95* is the 95th Percentile Error. *Acc.* is Top-1 Accuracy; *Bal. Acc.* is Balanced Accuracy. Best results are **bolded**; second best are underlined. **MFN (Gabor)** achieves the best balance of tasks. Notably, **ReLU+RFF** achieves the best high-frequency precision (Border *MedAE* < 4km) but suffers from severe global incoherence (high RMSE). **FINER** failed to converge.

Model	Best LR	Params (M)	Distance Regression (km)				Classification (%)				$S_{val}$
			RMSE (Global)	MedAE (Global)	P95AE (Global)	Border MedAE	Country ( $c_{in}$ )	Neighbor ( $c_{adj}$ )	Acc.	Bal. Acc.	
<b>MFN (Gabor)</b>	$4e^{-5}$	1.90	63.31	23.37	137.39	6.90	<b>99.3</b>	<u>84.3</u>	<b>98.6</b>	<b>92.7</b>	<b>86.29</b>
<b>HOSC</b>	$4e^{-4}$	1.88	<b>54.85</b>	22.02	<b>117.11</b>	6.05	98.8	78.5	97.3	87.4	<u>88.96</u>
<b>INCODE</b>	$1e^{-4}$	1.62	61.97	<b>20.57</b>	133.80	4.88	99.1	80.0	98.0	89.7	92.31
<b>SIREN</b>	$1e^{-5}$	1.88	67.19	<u>21.10</u>	146.62	<u>4.50</u>	99.2	82.4	98.3	90.2	94.59
<b>FR-SIREN</b>	$4e^{-5}$	1.88	<u>58.74</u>	25.12	<u>124.72</u>	6.62	98.7	75.4	97.1	86.3	97.10
<b>MFN (Fourier)</b>	$1e^{-4}$	1.89	101.12	24.93	218.23	5.81	<b>99.5</b>	<b>89.3</b>	<b>99.2</b>	<b>96.1</b>	115.77
<b>GAUSS</b>	$1e^{-4}$	1.88	81.77	28.56	175.92	6.18	98.8	76.5	97.1	87.6	117.70
<b>ReLU + PE</b>	$4e^{-4}$	2.66	91.91	24.03	194.23	5.60	98.9	78.2	97.5	89.9	123.78
<b>SINC</b>	$4e^{-4}$	1.88	78.87	31.14	165.17	6.95	98.0	70.2	95.3	82.4	126.36
<b>ReLU (Base)</b>	$4e^{-4}$	1.88	110.37	52.92	236.36	7.39	96.5	67.2	91.2	68.7	174.51
<b>ReLU + RFF</b>	$4e^{-4}$	2.14	153.98	38.19	335.19	<b>3.95</b>	99.1	82.9	97.6	88.1	182.99
<b>WIRE</b>	$4e^{-5}$	1.88	242.82	74.02	549.30	7.54	94.2	53.0	84.2	49.0	340.93
<b>FINER</b>	$1e^{-5}$	1.88	839.78	309.8	1935.5	154.2	0.1	0.4	6.1	0.4	1038.9



Fig. 1: **Political Classification (MFN-Fourier).** The model achieves near-perfect segmentation of countries (99.5% Accuracy).

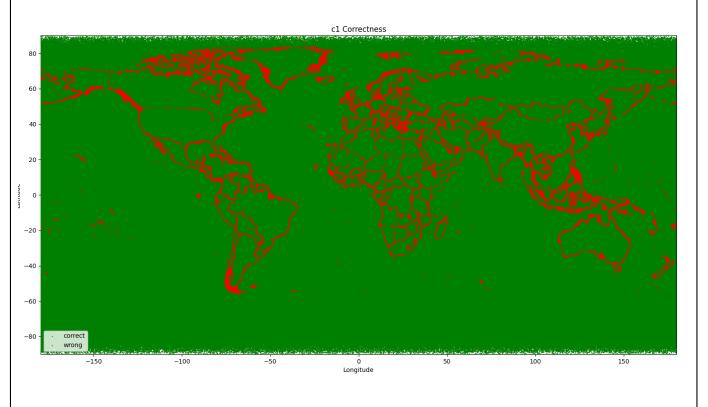


Fig. 2: **Political Classification Error (MFN-Fourier).** Errors (highlighted in red) are confined to immediate border regions, representing sub-pixel ambiguity rather than structural failure.

a true distance of 10km is penalized heavily, while a 10km error at a true distance of 5000km is negligible.

## 6. Discussion

Our results demonstrate a clear hierarchy among NIR architectures, with filter-based methods (MFN) significantly outperforming deep coordinate networks (SIREN, INCODE). However, the most salient finding is the collective inability of *any* global MLP architecture to satisfy the  $< 4\text{km}$  precision constraint within the 8MB budget while maintaining global coherence. This suggests a fundamental limitation in applying monolithic neural fields to planetary-scale geography.

### 6.1. The Spectral Wall of Global Geography

The core challenge lies in the extreme spectral heterogeneity of the signal. Political borders exhibit multi-scale geometry: the borders of Western Sahara are effectively lines (low frequency), while the coastlines of Norway are fractals (infinite frequency).

The "anti-aliasing" effect observed in Figure 3 perfectly illustrates the following: The network has sufficient capacity to locate the country roughly (low-frequency correctness) but lacks the spectral bandwidth to encode the jagged, high-frequency deviations of the actual border. Under the strict parameter budget, the optimization settles on a smooth approximation, a "blurred" border, rather than fitting the high-frequency noise which would consume the entire capacity of

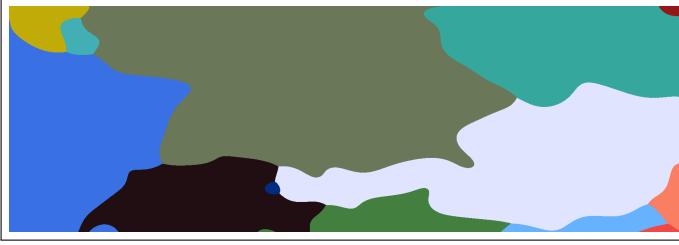


Fig. 3: **Political Classification around the Alpes (MFN-Fourier).** While the overall shapes of larger countries is preserved, the local borders are smooth, highlighting Spectral Bias.



Fig. 4: **Neighbor Classification (MFN-Fourier).** The model achieves near-perfect segmentation of nearest neighbor (99.2% Accuracy).

the network. We term this the **"Spectral Wall"**: a monolithic network cannot efficiently compress a signal where the local complexity varies by orders of magnitude across the domain.

## 6.2. Multi-Task Tension

We observed a competitive interference between the classification and regression tasks, evidenced by the performance gap between MFN-Fourier (best classifier) and MFN-Gabor (best overall).

- **Classification** requires a decision boundary that approaches a step function (infinite gradient).
- **Distance Regression** requires a smooth function ( $C^0$  continuous). Since we regress  $y = \ln(d + 1)$ , the target gradient magnitude is  $|\nabla y| = \frac{1}{d+1}$ , which requires the network to learn a decaying slope that is steepest exactly where the classification boundary is sharpest.

Architectures that mastered the sharp step function (MFN-Fourier, 99.5% accuracy) struggled to model the decaying gradient field, resulting in high RMSE (101 km). MFN-Gabor likely succeeded because the Gaussian windowing of the Gabor filters allows for "soft" localization, bridging the gap between discrete segmentation and continuous regression. The collective under-performance of all architectures suggests that for geospatial applications, disentangling these representations, perhaps using separate "geometry" and "semantic" heads earlier in the network trunk, may be necessary.

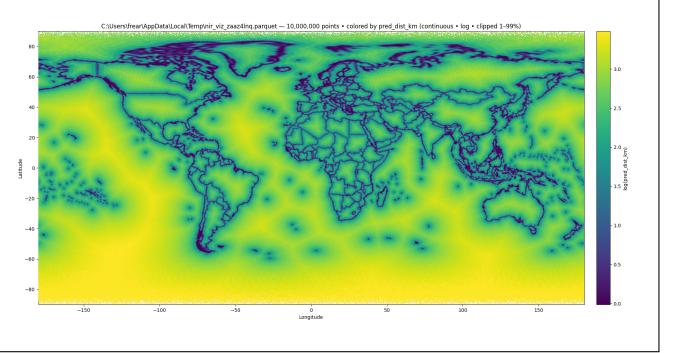


Fig. 5: **Distance Field Regression (MFN-Gabor).** The predicted distance field shows smooth gradients radiating from landmasses.

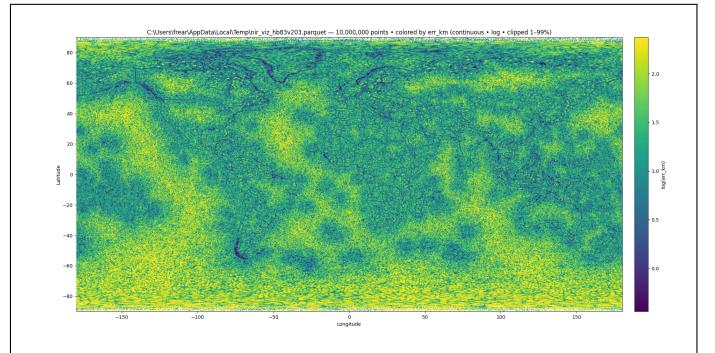


Fig. 6: **Distance Field Regression (MFN-Gabor).** The absolute error map (km) shows that error magnitude scales with distance, peaking in open oceans where precision is less critical for navigation.

## 6.3. Architecture Analysis

**Filter Networks vs. Deep Composition:** The success of **MFN (Gabor)** suggests that for spherical geometry, recursive filter application is more effective than standard deep composition. The Gabor filter, combining a sinusoidal carrier with a Gaussian window, allows the network to isolate specific geographic regions (e.g., an island) and apply high-frequency details locally without disturbing the global field. This effectively acts as a "soft" spatial partitioning.

**The Failure of Complex Activations:** FINER's divergence highlights the risk of coupling frequency modulation directly to input magnitude ( $\omega(|x| + 1)$ ) on a manifold where inputs are normalized unit vectors ( $|x| = 1$ ). This formulation likely introduced numerical instability in the gradients. Similarly, **WIRE** (complex Gabor) suffered from low classification accuracy, suggesting the complex-valued landscape was too difficult to traverse under our fixed budget.

**6.3.0.1. Inefficacy of Fourier Reparameterization (FR):** FR-SIREN operated within the same inference-time parameter budget ( $\approx 1.88M$ ) as the standard baseline but failed to provide a statistically significant improvement ( $S_{val}$  of 97.10 vs 94.59). This suggests that the bottleneck is not the optimization

landscape (which FR targets), but the capacity of the fixed-basis representation itself to encode multi-scale spherical data.

## 6.4. Limitations

While our benchmark provides a comprehensive overview of architectural capabilities, several limitations constrain the generalizability of our findings.

### 6.4.0.1. Architectural Hyperparameter Constraints:

While we performed a grid search over the global learning rate to ensure fair optimization, we adopted a *Standardized Capacity* approach, fixing network size to  $\approx 8\text{MB}$  and using default frequency parameters. We did not perform architecture-specific tuning (e.g., via Optuna) due to the computational cost of the 200M-sample training regime. Consequently, our results represent a robust lower bound on performance. However, given the magnitude of the error gap (Order of tens of kilometers), we argue that hyperparameter tuning alone is unlikely to bridge the deficit without structural changes.

### 6.4.0.2. Single-Scale Training:

We treated the Earth as a single manifold. In practice, production GIS systems use distinct Level-of-Detail (LOD) models. Our strict requirement for a *single* global network penalizes architectures that might otherwise excel as leaf-nodes in a hierarchical structure.

## 7. Conclusion

In this work, we investigated the feasibility of replacing explicit vector geometries with compact Neural Implicit Representations for global political geography. We benchmarked twelve architectures on a multi-task learning problem involving spherical distance fields and country classification.

Our experiments reveal three key findings:

- 1) **Filter Networks Outperform Deep Composition:** The Multiplicative Filter Network (MFN-Gabor) achieved the best aggregate performance ( $S_{\text{val}}$  of 86.29), outperforming deep coordinate networks like INCODE and SIREN. This suggests that recursive filter application is mathematically better suited for spherical signals than standard depth-based composition.
- 2) **The Precision-Coherence Trade-off:** We observe a Pareto frontier between local precision and global coherence. The **ReLU+RFF** model was the only architecture to satisfy the  $< 4\text{km}$  border precision requirement (3.95 km Border MedAE), but it exhibited severe global artifacts (153 km RMSE). Conversely, **HOSC** achieved the smoothest global field (54 km RMSE) but failed to capture sharp borders.
- 3) **The Spectral Wall:** No single global architecture could simultaneously satisfy the  $< 10\text{km}$  global error and  $< 4\text{km}$  border error constraints within the 8MB budget. Visual analysis confirms that errors are concentrated at high-frequency boundaries, manifesting as anti-aliased approximations of fractal coastlines.

**Future Work:** The failure of global MLPs to bridge the gap between microscopic borders and macroscopic oceans

suggests that the future of Neural GIS lies not in better global activations, but in **Spatial Decomposition**. Hybrid architectures such as **Neural Geometric Level of Detail (NGLOD)** [14] or ACORN [15], which utilize Octrees or Quadtrees to assign dedicated small MLPs to complex spatial regions, are the logical next step. By explicitly partitioning the sphere, a model could allocate dense resources to Europe's borders while using a sparse representation for the Pacific Ocean, breaking the "Spectral Wall" identified in this study.

## References

- [1] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.
- [2] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in neural information processing systems*, vol. 33, pp. 7537–7547, 2020.
- [3] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.
- [4] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 5301–5310.
- [5] Z. Liu, H. Zhu, Q. Zhang, J. Fu, W. Deng, Z. Ma, Y. Guo, and X. Cao, "Finer: Flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2713–2722.
- [6] R. Fathony, A. K. Sahu, D. Willmott, and J. Z. Kolter, "Multiplicative filter networks," in *International conference on learning representations*, 2020.
- [7] A. Kazerouni, R. Azad, A. Hosseini, D. Merhof, and U. Bagci, "Incode: Implicit neural conditioning with prior knowledge embeddings," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 1298–1307.
- [8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [9] D. Serrano, J. Szymkowiak, and P. Musalski, "Hosc: A periodic activation function for preserving sharp features in implicit neural representations," *arXiv preprint arXiv:2401.10967*, 2024.
- [10] H. Saratchandran, S. Ramasinghe, V. Shevchenko, A. Long, and S. Lucey, "A sampling theory perspective

- on activations for implicit neural representations,” *arXiv preprint arXiv:2402.05427*, 2024.
- [11] S. Ramasinghe and S. Lucey, “Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps,” in *European Conference on Computer Vision*. Springer, 2022, pp. 142–158.
  - [12] V. Saragadam, D. LeJeune, J. Tan, G. Balakrishnan, A. Veeraraghavan, and R. G. Baraniuk, “Wire: Wavelet implicit neural representations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 507–18 516.
  - [13] K. Shi, X. Zhou, and S. Gu, “Improved implicit neural representation with fourier reparameterized training,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 25 985–25 994.
  - [14] T. Takikawa, A. Evans, J. Tremblay, T. Müller, M. McGuire, A. Jacobson, and S. Fidler, “Variable bitrate neural fields,” in *ACM SIGGRAPH 2022 Conference Proceedings*, ser. SIGGRAPH ’22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: <https://doi.org/10.1145/3528233.3530727>
  - [15] J. N. Martel, D. B. Lindell, C. Z. Lin, E. R. Chan, M. Monteiro, and G. Wetzstein, “Acorn: Adaptive coordinate networks for neural scene representation,” *arXiv preprint arXiv:2105.02788*, 2021.

## 8. Appendix A: Architecture Implementation Details

This appendix details the specific initialization schemes and hyperparameter configurations used for the benchmarked architectures. All implementations rely on the PyTorch framework.

### 8.1. Initialization Regimes

8.1.0.1. SIREN Initialization: Following Sitzmann et al. [3], weights are initialized from a uniform distribution  $\mathcal{U}(-\text{bound}, \text{bound})$  to preserve activation distribution variance through the network.

- **First Layer:** bound =  $1/n_{in}$ , where  $n_{in}$  is the number of input features.
- **Hidden Layers:** bound =  $\sqrt{6/n_{in}}/\omega$ .
- **Biases:** In our implementation, biases for the SIREN layers are initialized using the default PyTorch initialization for linear layers.

8.1.0.2. FINER Initialization: FINER [5] modifies the bias initialization to separate neurons into different frequency bands.

- **Weights:** Initialized identically to SIREN hidden layers:  $\mathcal{U}(-\sqrt{6/n_{in}}/\omega, \sqrt{6/n_{in}}/\omega)$ .
- **Biases:** Initialized uniformly from  $\mathcal{U}(-k, k)$ . For this work, we use the authors' recommended value of  $k = 20$ .

8.1.0.3. MFN (Gabor) Initialization: The Gabor Multiplicative Filter Network [6] requires initializing the filter parameters (geometric centers  $\mu$  and bandwidths  $\gamma$ ) and the linear weights.

- **Centers ( $\mu$ ):** Initialized uniformly over the input domain:  $\mu \sim \mathcal{U}(-1, 1)$ .
- **Bandwidths ( $\gamma$ ):** Initialized from a Gamma distribution to ensure a mix of broad and sharp filters. We sample  $g \sim \Gamma(\alpha, \beta)$  where  $\alpha = 6.0/\text{depth}$  and  $\beta = 1.0$ . The raw parameter is set to the inverse softplus of  $g$  to ensure positivity during training.
- **Filter Linear Weights:** Initialized using the default PyTorch distribution but scaled by the network depth and filter bandwidth to maintain signal magnitude:

$$\mathbf{W} \leftarrow \mathbf{W} \times \frac{n_{out}}{\sqrt{\text{depth}}} \times \sqrt{\gamma} \quad (5)$$

- **Filter Biases:** Initialized uniformly  $\phi \sim \mathcal{U}(-\pi, \pi)$  to cover all phases.

8.1.0.4. Fourier Reparameterization (FR) Initialization: For FR-SIREN, the weights  $\mathbf{W}$  are decomposed into  $\mathbf{\Lambda}\mathbf{B}$ . The basis matrix  $\mathbf{B}$  is fixed, and only  $\mathbf{\Lambda}$  is learned.

- **Basis ( $\mathbf{B}$ ):** Composed of cosine functions sampled at frequencies  $\omega \in \{1/F, \dots, F\}$  and phases  $\phi \in [0, 2\pi]$ .
- **Coefficients ( $\mathbf{\Lambda}$ ):** Initialized to ensure the variance of the resulting effective weight  $\mathbf{W}$  matches standard initialization. For the  $j$ -th column, we sample  $\Lambda_{:,j} \sim \mathcal{U}(-b_j, b_j)$

where:

$$b_j = \frac{\sqrt{6/M}}{\|\mathbf{b}_j\|_2} \quad (6)$$

Here,  $M$  is the number of Fourier bases, and  $\|\mathbf{b}_j\|_2$  is the  $L_2$  norm of the  $j$ -th basis vector.

### 8.2. INCODE Regularization

To stabilize the dynamic frequency modulation in INCODE [7], the auxiliary Harmonizer network outputs  $(a, b, c, d)$  are regularized towards the neutral identity transform  $(1, 1, 0, 0)$ . We add the following term to the total loss:

$$\mathcal{L}_{\text{incode}} = \lambda_1(a - 1)^2 + \lambda_2(b - 1)^2 + \lambda_3c^2 + \lambda_4d^2 \quad (7)$$

We use the recommended coefficients:  $\lambda_1 = 0.1993$ ,  $\lambda_2 = 0.0196$ ,  $\lambda_3 = 0.0588$ , and  $\lambda_4 = 0.0269$ .

## 9. Appendix B: Data Engineering and Ground Truth Generation

To enable efficient querying of the exact signed distance field on the sphere, we implemented a custom high-performance geometric engine. This appendix details the data preprocessing and query pipeline.

### 9.1. Data Source and Preprocessing

The primary vector data is sourced from the World Bank ADM0 Global Boundaries dataset, containing 288 sovereign territories.

- 1) **Geometry Explode:** Multi-polygons are exploded into individual polygons. We extract the boundary linestrings and segmentize them into individual geodesic arcs.
- 2) **Coordinate Transformation:** All vertices are converted from Geodetic coordinates (Latitude  $\phi$ , Longitude  $\lambda$ ) to 3D Cartesian coordinates  $(x, y, z)$  on the unit sphere:

$$x = \cos \phi \cos \lambda, \quad y = \cos \phi \sin \lambda, \quad z = \sin \phi \quad (8)$$

This prevents singularities at the poles and allows for Euclidean distance metrics to approximate geodesic chords locally.

- 3) **Serialization:** The processed collection of  $\approx 10^7$  arc segments is serialized to a **FlatGeobuf** file (652 MB), enabling memory-mapped access during training.

### 9.2. Spatial Indexing (KDTree)

Calculating the exact distance from a query point to millions of border segments is computationally prohibitive ( $O(N)$ ). To accelerate this, we construct a **K-Dimensional Tree (KDTree)**:

- **Tree Construction:** We build the KDTree on the 3D Cartesian midpoints of every border segment.
- **Query Strategy:** For a query point  $\mathbf{x}$ , we query the KDTree for the  $k = 128$  nearest midpoints. This reduces

the search space from millions of segments to a localized subset.

- **Exact Distance:** We compute the exact point-to-segment distance for these  $k$  candidates. The minimum of these distances is taken as the ground truth geodesic distance  $d(\mathbf{x})$ .

### 9.3. Classification and Special Regions

9.3.0.1. Water and Antarctica:. The source dataset defines 288 land territories. To create a complete partition of the sphere  $\mathbb{S}^2$ , we define a 289th class,  $c_{water}$ , representing all regions not covered by the land polygons. Since the World Bank dataset excludes Antarctica, points sampled within the Antarctic region are implicitly classified as  $c_{water}$ . While this is a topological simplification, it ensures that every point on the sphere has a valid class label.

9.3.0.2. Point-in-Polygon Test:. To generate the ground truth classification label  $c_{in}(\mathbf{x})$ , we perform a spherical point-in-polygon test.

#### **9.4. Plagiarism statement**

I declare that I am aware of the following facts:

- I understand that in the following statement the term "person" represents a human or **ANY AUTOMATIC GENERATION SYSTEM**.
- As a student at the University of Luxembourg I must respect the rules of intellectual honesty, in particular not to resort to plagiarism, fraud or any other method that is illegal or contrary to scientific integrity.
- My report will be checked for plagiarism and if the plagiarism check is positive, an internal procedure will be started by my tutor. I am advised to request a pre-check by my tutor to avoid any issue.
- As declared in the assessment procedure of the University of Luxembourg, plagiarism is committed whenever the source of information used in an assignment, research report, paper or otherwise published/circulated piece of work is not properly acknowledged. In other words, plagiarism is the passing off as one's own the words, ideas or work of another person, without attribution to the author. The omission of such proper acknowledgement amounts to claiming authorship for the work of another person. Plagiarism is committed regardless of the language of the original work used. Plagiarism can be deliberate or accidental. Instances of plagiarism include, but are not limited to:
  - 1) Not putting quotation marks around a quote from another person's work
  - 2) Pretending to paraphrase while in fact quoting
  - 3) Citing incorrectly or incompletely
  - 4) Failing to cite the source of a quoted or paraphrased work
  - 5) Copying/reproducing sections of another person's work without acknowledging the source
  - 6) Paraphrasing another person's work without acknowledging the source
  - 7) Having another person write/author a work for oneself and submitting/publishing it (with permission, with or without compensation) in one's own name ('ghost-writing')
  - 8) Using another person's unpublished work without attribution and permission ('stealing')
  - 9) Presenting a piece of work as one's own that contains a high proportion of quoted/copied or paraphrased text (images, graphs, etc.), even if adequately referenced

Auto- or self-plagiarism, that is the reproduction of (portions of a) text previously written by the author without citing that text, i.e. passing previously authored text as new, may be regarded as fraud if deemed sufficiently severe.