mGPS Algorithm Optimization

Course: Bioinformatics Research Project (BINP37), 15 credits

Student: Chandrashekar CR

(email: ch1131ch-s@student.lu.se)

Supervisor: Eran Elhaik

(email: eran.elhaik@biol.lu.se)

Lund University 2025

Abstract

mGPS (microbiome Geographic Population Structure) is a novel algorithm developed to analyze the geographical distribution of microbial communities. This report presents an enhanced version of mGPS with significant improvements in computational efficiency and predictive accuracy, enabling robust analysis of large-scale datasets. Optimization efforts focused on refining the algorithm's core functionality, improving data handling, and implementing ensemble-based learning architectures.

The core of this study involves the application of advanced machine learning models to improve geolocation predictions based on microbiome signatures. The Metasub dataset, comprising 4,070 samples collected from 40 cities across seven continents between 2016 and 2017, served as the basis for evaluation. Several neural and ensemble models were developed and benchmarked against the original mGPS framework.

Our refined pipeline introduces a hierarchical ensemble learning architecture incorporating XGBoost, CatBoost, TabPFN, neural networks, and GrowNets, with performance-gated meta-models at each geographic resolution. Class imbalance was addressed using SMOTE, and stratified 5-fold cross-validation was employed to ensure fair and robust evaluation. A critical advancement includes a corrected error calculation methodology that more accurately quantifies performance by accounting for hierarchical prediction dependencies.

The optimized mGPS framework offers superior predictive performance and sets a foundation for future applications in public health surveillance, forensic investigation, and ecological research.

1. Introduction

1.1 Geographical Prediction Using Microbial Signatures

Microorganisms in environmental samples serve as biological signatures reflecting local environmental conditions, human activity patterns, and ecological factors unique to specific geographical regions [REF NEEDED]. This capability has significant implications for biosurveillance, forensic investigation, and public health monitoring [REF NEEDED].

The microbial Global Population Structure (mGPS) algorithm leverages these signatures for geographical prediction by analyzing relative sequence abundance (RSA) of microorganisms [REF NEEDED]. The original implementation employed a hierarchical XGBoost model with continent classification preceding city prediction, followed by coordinate inference, achieving 92% city-level accuracy and 137 km median error distance on the MetaSUB dataset [REF NEEDED].

1.2 Previous Work and Methodology

Several studies have built upon the mGPS framework using XGBoost with various improvements including hyperparameter optimization [REF NEEDED]. The standard workflow involves:

- Feature reduction from thousands to hundreds of informative microbial features via recursive feature elimination (RFE)
- Hierarchical prediction: continent → city → coordinates
- Augmentation of prediction probabilities at each level to enhance subsequent predictions

1.3 Limitations in Existing Approaches

Current methodologies exhibit critical limitations. Error metrics are interdependent—distance calculations benefit from high continent/city accuracy, potentially obscuring true performance [REF NEEDED]. The hierarchical structure propagates errors through prediction levels, with early misclassifications causing substantial geographical displacement in final coordinates [REF NEEDED].

Previous approaches inadequately address:

- Mathematical frameworks for quantifying hierarchical error propagation
- Dataset imbalances across geographical scales biasing model training
- Ensemble approaches for minimizing cascading errors
- Proper coordinate transformation for machine learning applications [REF NEEDED]

1.4 Research Objectives and Contributions

This research addresses: How can error propagation be minimized in hierarchical geographical prediction? What algorithmic combinations optimize microbial-based geographical prediction? How should error metrics reflect real-world performance in cascaded systems?

Our contributions include:

- Ensemble learning methodology combining multiple algorithms to minimize error propagation
- Mathematical framework for quantifying hierarchical prediction errors
- Enhanced coordinate transformation techniques for geospatial accuracy
- Comprehensive evaluation accounting for cascaded prediction errors

1.5 Dataset and Proposed Enhancements

This research utilizes the MetaSUB dataset: 4,070 quality-controlled samples from 40 cities across 7 continents [REF-metasub-2020]. Each sample contains taxonomic profiles with relative sequence abundances, initially comprising over 3,000 organisms, reduced to 200-300 informative features through RFE.

Our methodological improvements include:

- **Ensemble Framework**: Combining XGBoost, CatBoost, TabPFN, neural networks, and GrowNets with threshold-filtering for high-confidence predictions
- **Balanced Training**: SMOTE implementation to address geographical imbalances
- **Corrected Error Metrics**: Accurate performance measurement across hierarchical chains
- **Coordinate Optimization**: Machine learning-specific transformations for geographical accuracy

1.6 Paper Organization

Materials and Methods details dataset preparation, algorithms, and evaluation framework. Results presents accuracy findings comparing ensemble versus previous methods. Discussion examines implications for microbial forensics and environmental monitoring. Conclusion summarizes contributions and future directions.

2. Materials and Methods

2.1 Dataset and Preprocessing

This study utilized the MetaSUB dataset from the original mGPS study [REF], accessed through their GitHub repository. The dataset comprises 4,070 quality-controlled samples from subway stations across 40 cities on 7 continents, collected between 2016-2017. Each sample contains taxonomic profiles with relative sequence abundances computed after subsampling to 100,000 classified reads, generated using KrakenUniq based on the NCBI/RefSeq Microbial database.

For methodological consistency with previous mGPS studies, we applied identical quality control procedures [REF]: removal of cities with fewer than eight samples and recursive feature elimination (RFE) using Random Forest to reduce 3,000 microbial features to approximately 200-300 most informative features with 5-fold cross-validation. To address class imbalance, particularly for underrepresented continents like Oceania and Africa, we employed Synthetic Minority Over-sampling Technique (SMOTE) to achieve a 1:3 ratio between minority and majority classes [REF].

2.2 Model Development

We developed multiple modeling approaches to tackle the hierarchical geographic prediction problem, each with distinct advantages and characteristics.

2.2.1 Neural Networks

Separate Neural Network Models Inspired by the hierarchical approach in the original mGPS study, which utilized XGBoost [REF], we developed a set of independent neural networks to serve as baselines and to analyze error propagation at each prediction level. Specifically, we constructed three specialized models: (1) a Continent Network that predicts continent labels from microbial features; (2) a City Network that incorporates both microbial features and continent probabilities to predict city labels; and (3) a Coordinate Network that leverages microbial features, continent, and city probabilities to perform coordinate regression.

Default parameters and the hyperparameter search space for these models are provided in Supplementary Tables 18 and 19.

Each network architecture incorporates progressive dropout, batch normalization, and ReLU activation functions. For coordinate prediction, we employ a 3D Cartesian transformation to appropriately model the spherical geometry of the Earth.

Each neural network in the separate hierarchy is trained independently using a standard loss function for its task:

- **Continent and City Classification:** Cross-entropy loss is used for both continent and city classification tasks. Class weights are optionally applied to address class imbalance:

$$\mathcal{L}_{\text{classification}} = \text{CrossEntropyLoss}(\text{predictions}, \text{targets}, \text{weight} = w_{\text{class}})$$

- **Coordinate Regression:** Mean squared error (MSE) loss is used for coordinate regression:

$$\mathcal{L}_{\text{regression}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2$$

Each model is trained independently with its respective loss function, and no explicit weighting between tasks is used in this separate approach.

Table 1. Architecture and training parameters for separate neural networks.

Level	Task	Hidden Layers	Dropout	Batch Norm	Learning Rate	Batch Size	Epochs
1	Continent	[128, 64]	0.3–0.7	Yes	1×10^{-3}	128	400
2	City	[256, 128, 64]	0.3–0.7	Yes	1×10^{-3}	128	400
3	Coordinates	[256, 128, 64]	0.2–0.5	Yes	1×10^{-4}	64	600

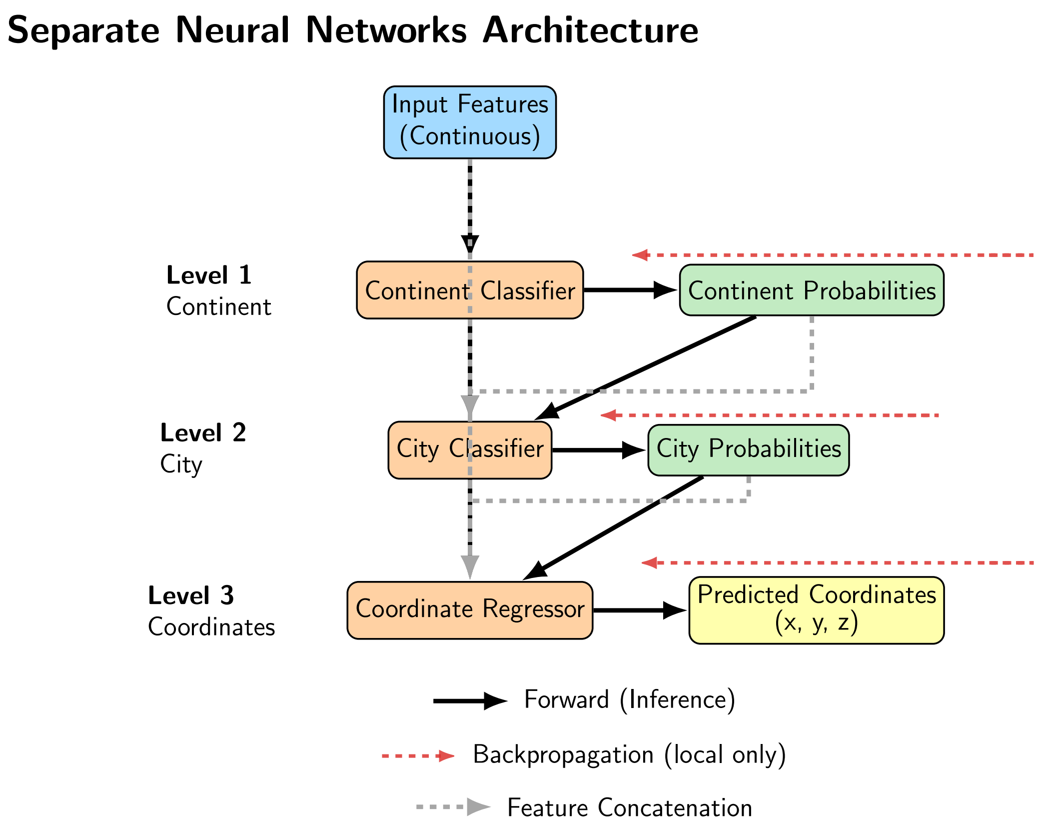

Figure 1. Schematic of the separate neural network approach for hierarchical geographic prediction. Each prediction level (continent, city, coordinates) is modeled by an independent neural network. Outputs from each level are used as inputs for the next, but training and backpropagation are performed independently for each network.

For each prediction level, the loss function is computed and backpropagated independently, ensuring that parameter updates for continent, city, and coordinate models remain decoupled.

Combined Hierarchical Neural Networks To enable end-to-end hierarchical learning, we developed the CombinedHierarchicalNet, a unified multi-task neural network architecture with three sequential branches. This model shares feature representations across tasks while maintaining task-specific output heads. Training is performed using a weighted multi-task loss, combining cross-entropy for classification tasks and mean squared error (MSE) for coordinate regression.

Default parameters and the hyperparameter search space for CombinedHierarchicalNet are provided in Supplementary Tables 20 and 21. City-level results are in Supplementary Table **??**, and in-radius accuracy breakdowns in Supplementary Tables 36 and **??**.

A key innovation of this architecture is the use of task-specific loss weights, where continent classification is assigned the highest weight, followed by city classification, and finally coordinate regression. This weighting scheme reflects the hierarchical structure of the problem, penalizing errors at higher levels more strongly to mitigate error propagation.

During backpropagation, gradients flow through all branches, but their magnitudes are modulated by these weights, promoting robust feature learning across the hierarchy.

Table 2. Architecture and training parameters for CombinedHierarchicalNet.

Branch	**Hidden Layers**	**Dropout**	**Batch Norm**	**Loss**	**Learning Rate**
Continent	[128, 64]	0.3–0.7	Yes	Cross-entropy	1×10^{-3}
City	[256, 128, 64]	0.3–0.7	Yes	Cross-entropy	1×10^{-3}
Coordinates	[256, 128, 64]	0.2–0.5	Yes	MSE	1×10^{-3}

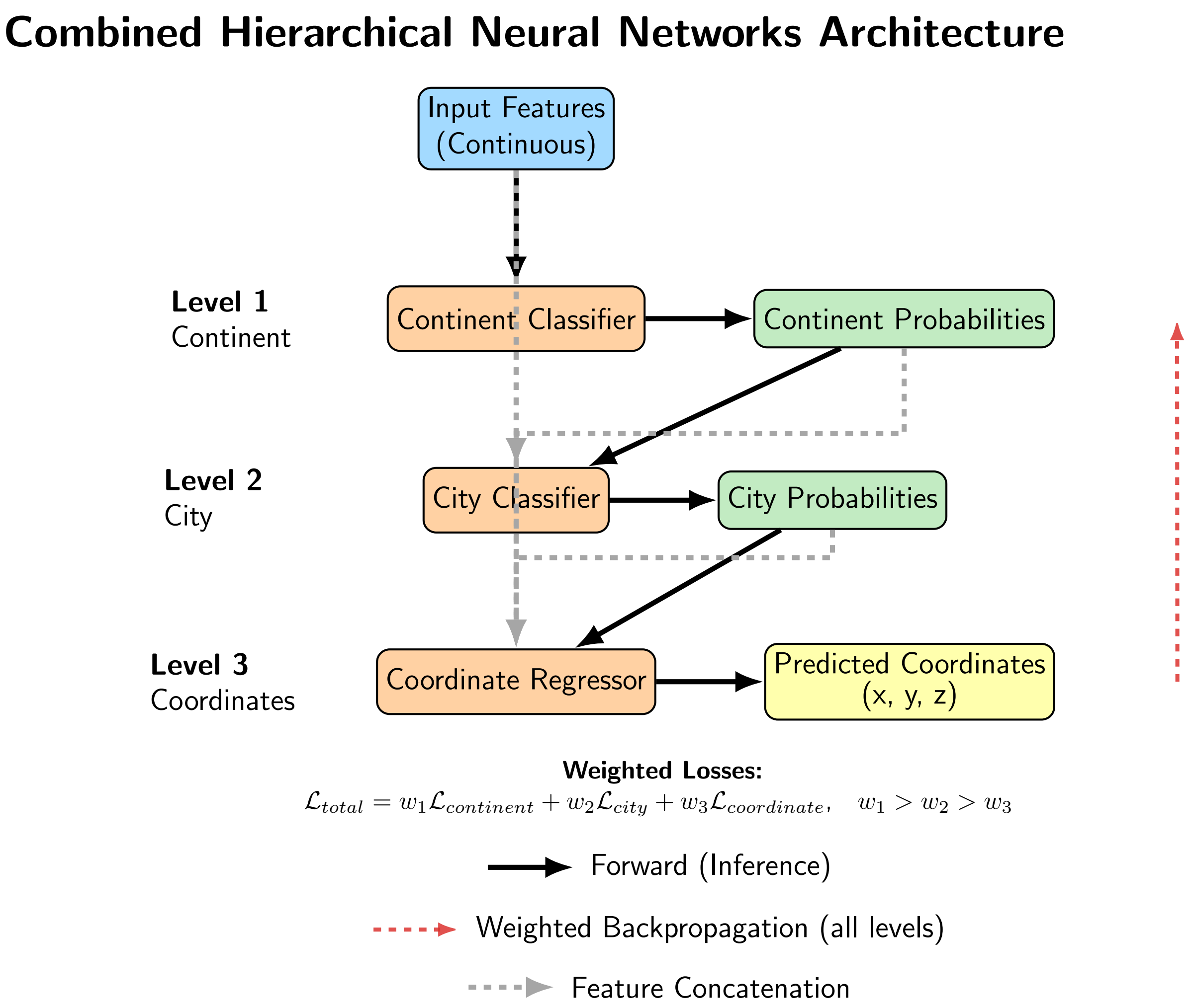

Figure 2. Diagram of the CombinedHierarchicalNet architecture. This unified multi-task neural network consists of sequential branches for continent, city, and coordinate prediction. Feature representations are shared, and predictions from higher levels are concatenated with features for downstream tasks. Training uses a weighted multi-task loss to reflect the hierarchy.

The total loss is defined as:

$$\mathcal{L}_{total} = w_1\mathcal{L}_{continent} + w_2\mathcal{L}_{city} + w_3\mathcal{L}_{coordinate} \tag{1}$$

where w_1, w_2, w_3 are the task-specific weights. This joint optimization strategy encourages the model to learn representations that are robust to error propagation, outperforming separate networks in empirical evaluations [REF].

2.2.2 GrowNet Architecture

We further evaluated GrowNet, a gradient boosting framework that employs neural networks as weak learners for multi-task learning [REF]. GrowNet sequentially adds shallow neural networks to the ensemble, each trained to correct the residuals of the previous learners, analogous to boosting in XGBoost.

Default parameters and the hyperparameter search space for GrowNet are provided in Supplementary Tables 22 and 23. City-level results are in Supplementary Table **??**, and in-radius accuracy breakdowns in Supplementary Tables 37 and **??**.

The hierarchical GrowNet training algorithm proceeds as follows:

1. **Input:** Training data $\{(\mathbf{x}_i, \mathbf{y}_{c,i}, \mathbf{y}_{city,i}, \mathbf{y}_{coord,i})\}_{i=1}^{N}$, hyperparameters M (number of stages), ρ (learning rate), λ (optimizer step size), and epochs_per_stage.
2. Initialize baseline predictions $F^{(0)}$.
3. For $m = 1$ to M:
   (a) Compute pseudo-residuals $\mathbf{r}^{(m)}$.
   (b) Initialize a new weak learner h_m.
   (c) For each epoch in epochs_per_stage:
      i. Sample a mini-batch B.
      ii. Compute gradients and update h_m parameters using $\nabla_\theta \mathcal{L}_{residual}(B; h_m)$.
   (d) Update ensemble: $F^{(m)} = F^{(m-1)} + \rho \cdot h_m$.
   (e) Periodically, jointly fine-tune all weak learners via corrective optimization:
   $$\{\theta_1, \ldots, \theta_m\} \leftarrow \arg\min_{\{\theta_i\}} \mathcal{L}_{total}(F^{(m)}; \{\theta_i\}_{i=1}^{m}) \quad (2)$$
   (f) Evaluate on validation data and apply early stopping if necessary.
4. Return the final ensemble $\mathcal{F} = \{h_1, \ldots, h_M\}$.

This corrective optimization step enables earlier weak learners to adapt based on information acquired by subsequent learners, enhancing ensemble coherence and predictive performance.

The hierarchical GrowNet model uses a composite loss function that combines classification and regression objectives at three levels: continent, city, and coordinates. The total loss is computed as a weighted sum of the individual task losses:

$$\mathcal{L}_{\text{total}} = w_1 \cdot \mathcal{L}_{\text{continent}} + w_2 \cdot \mathcal{L}_{\text{city}} + w_3 \cdot \mathcal{L}_{\text{coordinate}}$$

where:

- $\mathcal{L}_{\text{continent}}$ is the cross-entropy loss for continent classification (optionally with class weights),
- $\mathcal{L}_{\text{city}}$ is the cross-entropy loss for city classification (optionally with class weights),
- $\mathcal{L}_{\text{coordinate}}$ is the mean squared error (MSE) loss for coordinate regression,
- w_1, w_2, w_3 are task-specific weights (typically $w_1 > w_2 > w_3$).

This loss encourages accurate predictions at each hierarchy level while allowing the user to emphasize higher-level tasks (e.g., continent) to mitigate error propagation.

2.2.3 Ensemble Learning

Building on insights from individual models, we designed a hierarchical ensemble framework that integrates complementary algorithmic strengths while minimizing error propagation.

Default parameters and the hyperparameter search space for the ensemble meta-models and all base models are provided in Supplementary Tables 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, and 34. City-level results are in Supplementary Table **??**, and in-radius accuracy breakdowns in Supplementary Tables 38 and **??**.

Model Selection and Integration The ensemble incorporates the following model families:

- **Gradient Boosting Models:** XGBoost (see Supplementary Tables 24, 25), LightGBM (see Supplementary Tables 26, 27), and CatBoost (see Supplementary Tables 28, 29), which are highly effective for capturing non-linear relationships in tabular data.
- **TabPFN:** A state-of-the-art prior-data fitted neural network specifically designed for small-to-medium tabular datasets [REF] (see Supplementary Table 34). TabPFN leverages meta-learning to rapidly adapt to new tasks, making it particularly suitable for our problem setting.

- **Neural Networks:** Standard multilayer perceptrons (MLPs) and hierarchical variants (see Supplementary Tables 32, 33), included for their capacity to model complex feature interactions, especially as dataset size increases.
- **GrowNet:** The aforementioned gradient boosting neural network architecture (see Supplementary Tables 30, 31), included as a robust alternative for scenarios with larger datasets or more intricate relationships.

Machine learning models were prioritized due to their strong empirical performance on tabular datasets. Neural network-based models and GrowNet were included as flexible alternatives for scenarios requiring greater model capacity.

Hierarchical Ensemble Architecture The ensemble is structured into three layers, each corresponding to a level in the geographic hierarchy:

Layer 1: Continent Classification Multiple base models predict continent probabilities from microbial features. Models are filtered based on cross-validation accuracy (threshold: 93%). SMOTE is applied to address class imbalance. Retained models generate out-of-fold predictions via 5-fold cross-validation, which are then used as meta-features for an XGBoost meta-model.

Layer 2: City Classification City prediction utilizes both the original microbial features and continent probability outputs from Layer 1. Models surpassing a 91% accuracy threshold are included in meta-learning, following the same protocol as Layer 1.

Layer 3: Coordinate Prediction Coordinate prediction leverages the full feature set: microbial abundances, continent probabilities, and city probabilities. Two approaches are considered:

- **Tree-based Models:** Latitude is predicted first, followed by longitude conditioned on the predicted latitude.
- **Neural Networks:** Direct prediction of 3D Cartesian coordinates, which are subsequently converted to latitude and longitude.

The model with the lowest median Haversine distance error is selected for final predictions; no meta-model is used at this stage.

Training Protocol and Meta-Learning Ensemble training proceeds in three stages:

- **Stage 1: Model Filtering and Meta-Feature Generation.** All base models undergo 5-fold stratified cross-validation to generate out-of-fold predictions. Only models meeting predefined performance thresholds are retained for meta-learning.

Table 3. Ensemble layer specifications and selection criteria.

Layer	Input Features	Selection Threshold	Meta-Model
Continent	Microbial (200-300)	93% accuracy	XGBoost
City	Microbial + continent probabilities	91% accuracy	XGBoost
Coordinates	Microbial + all probabilities	Best median distance	None

- **Stage 2: Hyperparameter Optimization.** Retained models are further optimized using Bayesian optimization (Optuna [REF]) with model-specific search spaces.
- **Stage 3: Meta-Model Training.** XGBoost meta-models are trained on concatenated probability outputs from the selected base models, enabling a learned ensemble strategy that outperforms simple averaging.

Table 4. Meta-model configuration parameters.

Parameter	Continent Meta-Model	City Meta-Model
Algorithm	XGBoost	XGBoost
Objective	Multi-class log-loss	Multi-class log-loss
Max depth	3	4
Learning rate	0.1	0.1
N-estimators	100	150
Subsample	0.8	0.8
Colsample bytree	0.8	0.8

Feature Augmentation and Data Flow The hierarchical ensemble implements systematic feature augmentation at each stage:

$$X_{cont} = \text{RFE}(X_{microbial}) \tag{3}$$

$$\hat{P}_{cont} = \text{MetaModel}_{cont}(\{f_i(X_{cont})\}_{i=1}^{N}) \tag{4}$$

$$X_{city} = [X_{cont}; \hat{P}_{cont}] \tag{5}$$

$$\hat{P}_{city} = \text{MetaModel}_{city}(\{f_j(X_{city})\}_{j=1}^{M}) \tag{6}$$

$$X_{coord} = [X_{cont}; \hat{P}_{cont}; \hat{P}_{city}] \tag{7}$$

$$\hat{Y}_{coord} = f_{best}(X_{coord}) \tag{8}$$

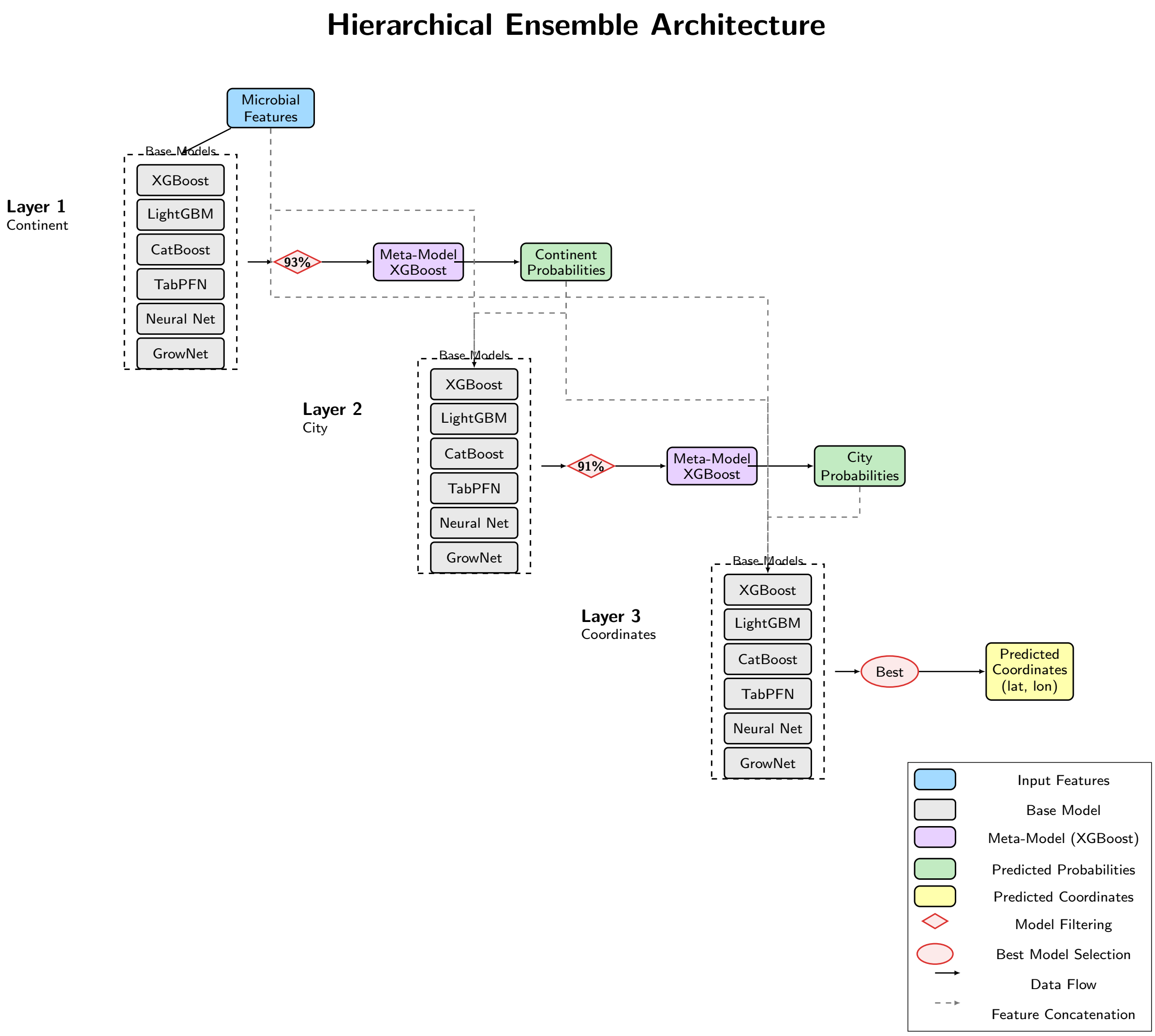

Figure 3. Overview of the hierarchical ensemble learning workflow. The ensemble is organized in three layers: continent classification, city classification, and coordinate regression. At each stage, predictions from multiple base models are combined using meta-models, and probability outputs are used as augmented features for subsequent layers.

3. Results

3.1 Neural Networks

3.1.1 Separate Neural Networks

The hierarchical neural network approach was evaluated in three stages: continent classification, city classification, and coordinate regression. The model was trained on 2604 samples and validated on 652 samples, with 814 samples in the final test set.

Level 1: Continent Classification The model achieved a validation accuracy of 84.9% for continent prediction. The macro-averaged F1-score was 0.78, indicating robust performance across all classes, despite some class imbalance. Table 5 summarizes the classification metrics.

Table 5. Continent Classification Report (Separate Neural Network)

Continent	Precision	Recall	F1-score	Support
east_asia	0.93	0.89	0.91	278
europe	0.86	0.82	0.84	283
middle_east	0.93	0.93	0.93	15
north_america	0.74	0.85	0.79	149
oceania	0.31	0.44	0.36	9
south_america	0.75	0.71	0.73	21
sub_saharan_africa	0.88	0.88	0.88	59
Accuracy		0.85 (814 samples)		
Macro avg	0.77	0.79	0.78	814
Weighted avg	0.86	0.85	0.85	814

Level 2: City Classification City-level classification yielded a validation accuracy of 70.1% (macro F1-score: 0.55), reflecting the increased difficulty and class imbalance at the city level. Detailed per-city metrics are provided in Supplementary Table **??**.

Level 3: Coordinate Regression The coordinate regression model achieved an RMSE of 0.581, MAE of 0.276, and R^2 of 0.658 on the test set (Table 6). Geodesic error analysis revealed a median error of 4237 km, mean error of 4962 km, and a maximum error of 17788 km. The expected coordinate error $E[D]$ was 4962 km. Table 7 details error breakdown by prediction correctness.

Table 6. Coordinate Regression Metrics (Separate Neural Network)

Metric	Value
Mean Squared Error (MSE)	0.338
Mean Absolute Error (MAE)	0.276
Root Mean Squared Error (RMSE)	0.581
R^2 Score	0.658

Table 7. Error Group Analysis (Separate Neural Network)

Group	Count	Mean Error (km)	Median Error (km)	Proportion	Weighted Error
C_correct Z_correct	565	3994	3255	0.694	2772
C_correct Z_wrong	126	5333	3703	0.155	826
C_wrong Z_correct	6	7668	8555	0.007	57
C_wrong Z_wrong	117	9098	7532	0.144	1308

The proportion of predictions within various geodesic radii is shown in Table 8. Notably, only 1.8% of predictions were within 1000 km, and 55.7% within 5000 km, highlighting the challenge of fine-grained localization.

Table 8. In-Radius Accuracy Metrics (Separate Neural Network)

Radius	Proportion (%)
<1 km	0.00
<5 km	0.00
<50 km	0.00
<100 km	0.00
<250 km	0.00
<500 km	0.86
<1000 km	1.84
<5000 km	55.65

Per-continent and per-city in-radius accuracy metrics are provided in Supplementary Tables 35 and **??**.

The hierarchical neural network achieved strong continent classification, moderate city classification, and limited coordinate precision. Most predictions were within continental scale, but city and coordinate errors remained substantial.

3.1.2 Combined Neural Networks

The combined hierarchical neural network jointly predicts continent, city, and coordinates. On the test set, it achieved 82.7% continent accuracy, 74.9% city accuracy, and coordinate RMSE of 0.237 (MAE: 0.126, R^2: 0.699).

Continent Classification Table 9 summarizes continent-level metrics. Macro F1-score was 0.75.

Table 9. Continent Classification Report (Combined Neural Network)

Continent	Precision	Recall	F1-score	Support
east_asia	0.90	0.90	0.90	278
europe	0.89	0.74	0.81	283
middle_east	0.70	0.93	0.80	15
north_america	0.72	0.85	0.78	149
oceania	0.33	0.44	0.38	9
south_america	0.65	0.81	0.72	21
sub_saharan_africa	0.80	0.90	0.85	59
Accuracy	0.83 (814 samples)			
Macro avg	0.71	0.80	0.75	814
Weighted avg	0.84	0.83	0.83	814

City Classification City-level accuracy was 74.9% (macro F1-score: 0.45). Full per-city results are provided in Supplementary Table **??**.

Coordinate Regression Coordinate regression achieved RMSE 0.237, MAE 0.126, and R^2 0.699. The median geodesic error was 519 km, mean 1631 km, and maximum 19604 km. Table 10 details error breakdown.

Table 10. Error Group Analysis (Combined Neural Network)

Group	Count	Mean Error (km)	Median Error (km)	Proportion	Weighted Error
C_correct Z_correct	581	502	274	0.714	358
C_correct Z_wrong	92	2101	1523	0.113	237
C_wrong Z_correct	29	3434	2252	0.036	122
C_wrong Z_wrong	112	6637	5377	0.138	913

In-radius accuracy improved substantially: 66.3% of predictions were within 1000 km, and 89.3% within 5000 km (Table 11).

Per-continent and per-city in-radius accuracy metrics are provided in Supplementary Tables 36 and **??**.

The combined neural network improved city and coordinate accuracy over the separate model, with a substantial reduction in geodesic error.

Table 11. In-Radius Accuracy Metrics (Combined Neural Network)

Radius	Proportion (%)
<1 km	0.00
<5 km	0.00
<50 km	0.37
<100 km	9.46
<250 km	30.34
<500 km	49.75
<1000 km	66.34
<5000 km	89.31

3.2 Hierarchical GrowNet

GrowNet achieved the highest continent (86.4%) and city (75.1%) classification accuracy among neural models, with coordinate MSE of 0.318. Table 12 summarizes continent metrics.

Table 12. Continent Classification Report (GrowNet)

Continent	Precision	Recall	F1-score	Support
east_asia	0.94	0.94	0.94	278
europe	0.87	0.81	0.84	283
middle_east	0.70	0.93	0.80	15
north_america	0.75	0.87	0.80	149
oceania	0.29	0.22	0.25	9
south_america	1.00	0.81	0.89	21
sub_saharan_africa	0.89	0.85	0.87	59
Accuracy	0.86 (814 samples)			
Macro avg	0.78	0.78	0.77	814
Weighted avg	0.87	0.86	0.86	814

City-level results (accuracy: 75.1%, macro F1: 0.60) are detailed in Supplementary Table ??.

Coordinate regression yielded a median geodesic error of 823 km, mean 1885 km, and maximum 18964 km. In-radius accuracy: 57.4% within 1000 km, 89.1% within 5000 km.

Per-continent and per-city in-radius accuracy metrics are provided in Supplementary Tables 37 and ??.

GrowNet outperformed all neural models in classification but not in coordinate regression.

Table 13. Error Group Analysis (GrowNet)

Group	Count	Mean Error (km)	Median Error (km)	Proportion	Weighted Error
C_correct Z_correct	604	904	599	0.742	671
C_correct Z_wrong	99	2215	1710	0.122	269
C_wrong Z_correct	7	4501	4324	0.009	39
C_wrong Z_wrong	104	7090	5896	0.128	906

3.3 Ensemble Learning

The ensemble approach, using XGBoost, LightGBM, and TabPFN, achieved state-of-the-art results. XGBoost and LightGBM reached 94.8% continent accuracy and 91.1% city accuracy (see Supplementary Table **??**). TabPFN achieved the best coordinate regression, with a test median distance of 13.9 km and mean of 581.5 km.

Table 14. Continent Classification Report (Ensemble, XGBoost)

Continent	Precision	Recall	F1-score	Support
east_asia	0.95	0.97	0.96	278
europe	0.95	0.94	0.95	283
middle_east	0.93	0.93	0.93	15
north_america	0.93	0.97	0.95	149
oceania	0.67	0.44	0.53	9
south_america	1.00	0.86	0.92	21
sub_saharan_africa	0.98	0.95	0.97	59
Accuracy		0.95 (814 samples)		
Macro avg	0.92	0.87	0.89	814
Weighted avg	0.95	0.95	0.95	814

Continent Classification

City Prediction For city classification, the ensemble achieved strong results, with XGBoost and LightGBM both exceeding 91% accuracy in cross-validation. The final meta-model achieved a test accuracy of 91.1%.

Coordinate Regression and Geodesic Error For coordinate regression, the ensemble leveraged TabPFN, which achieved the best geodesic performance. The test set median distance error was 13.9 km, with a mean distance error of 581.5 km and a 95th percentile error of 3703.5 km. The expected coordinate error, weighted by prediction correctness, was 581.5 km. Table ?? summarizes the error breakdown by prediction correctness.

Table 15. Ensemble: Error Group Analysis (TabPFN, test set)

Group	Count	Mean Error (km)	Median Error (km)	Proportion	Weighted Error
C_correct Z_correct	728	179.1	11.7	0.894	160.2
C_correct Z_wrong	44	2201.8	2086.4	0.054	119.0
C_wrong Z_correct	16	3964.9	3430.9	0.020	77.9
C_wrong Z_wrong	26	7026.0	6766.4	0.032	224.4

In-Radius Accuracy Table ?? summarizes the proportion of predictions within various geodesic radii. The ensemble model achieves 86.7% of predictions within 1000 km and 96.6% within 5000 km of the true location, substantially outperforming all neural network-based models.

Table 16. Ensemble: In-Radius Accuracy Metrics (TabPFN, test set)

Radius	Proportion (%)
<1 km	0.00
<5 km	4.67
<50 km	67.44
<100 km	71.74
<250 km	78.13
<500 km	82.19
<1000 km	86.73
<5000 km	96.56

4. Comparison with Previous State-of-the-Art (mGPS)

The mGPS (microbiome geographic population structure) tool represents the previous state-of-the-art for predicting the geographical origins of metagenomic samples from the MetaSUB dataset. Table 17 summarizes the key comparable metrics between mGPS and our ensemble model.

Summary The ensemble model achieves comparable city-level accuracy and substantially improved coordinate precision (lower median error, higher in-radius accuracy) relative to mGPS. Additional metrics (AUC, AUPR, sensitivity, specificity, and fine-scale within-city performance) will be computed for a more comprehensive comparison in future work. precision (lower median error, higher in-radius accuracy) relative to mGPS.

Table 17. Comparison of Ensemble Model and mGPS on MetaSUB Dataset

Metric	mGPS	Ensemble (TabPFN)	Notes	Reference
Sample Size	4,070 (40 cities)	4,070 (40 cities)	After QC, matched setup	–
City Prediction Accuracy	92%	91.1%	Test set	Table **??**
Sensitivity	78%	–	To be computed	–
Specificity	99%	–	To be computed	–
In-Radius Accuracy				
<250 km	62%	78.1%	–	Table 16
<500 km	74%	82.2%	–	Table 16
<1,000 km	84%	86.7%	–	Table 16
Median Error (km)	137	13.9	–	Table 15
AUC (Continent/City)	0.99–0.996	–	To be computed	–
AUPR (Continent/City)	0.97 / 0.87	–	To be computed	–
Fine-Scale (Within-City)				
Hong Kong (station accuracy)	82%	–	Not yet computed	–
Hong Kong (median error)	1.25 km	–	Not yet computed	–
New York (station/borough)	43% / 64%	–	Not yet computed	–
New York (median error)	2.39 km	–	Not yet computed	–
London (region accuracy)	48%	–	Not yet computed	–
AMR Tracing, GITs, Temporal Robustness	Demonstrated	–	Not evaluated	–

Additional metrics (AUC, AUPR, sensitivity, specificity, and fine-scale within-city performance) will be computed for a more comprehensive comparison in future work.

5. Discussion

[Discuss your results here. Interpret your findings, compare with previous studies, discuss limitations, and suggest future directions.]

6. Supplementary Materials

6.1 Separate Neural Network Parameters

Table 18. Default parameters for separate neural network models

Parameter	Continent Model	City Model	Coordinate Model
Hidden dimensions	[128, 64]	[256, 128, 64]	[256, 128, 64]
Batch normalization	True	True	True
Initial dropout	0.3	0.3	0.2
Final dropout	0.7	0.7	0.5
Learning rate	1e-3	1e-3	1e-4
Weight decay	1e-5	1e-5	1e-5
Batch size	128	128	64
Epochs	400	400	600
Early stopping steps	20	20	30
Gradient clip	1.0	1.0	1.0

Table 19. Hyperparameter search space for neural network tuning

Hyperparameter	Search Space
Hidden dimensions	[64], [128], [128, 64], [256, 128, 64], [256, 128], [512, 256, 128, 64]
Initial dropout	0.1 to 0.3
Final dropout	0.5 to 0.8
Learning rate	1e-4 to 1e-2 (log uniform)
Batch size	64, 128, 256
Weight decay	1e-6 to 1e-3 (log uniform)
Gradient clip	0.5 to 2.0

6.2 Combined Neural Network Parameters

Table 20. Default parameters for combined neural network model

Parameter	**Value**
Architecture parameters	
Continent branch hidden dimensions	[128, 64]
City branch hidden dimensions	[256, 128, 64]
Coordinate branch hidden dimensions	[256, 128, 64]
Continent branch dropout (initial, final)	(0.3, 0.7)
City branch dropout (initial, final)	(0.3, 0.7)
Coordinate branch dropout (initial, final)	(0.2, 0.5)
Batch normalization	True
Training parameters	
Learning rate	1e-3
Weight decay	1e-5
Batch size	128
Epochs	600
Early stopping steps	50
Continent loss weight	1.0
City loss weight	0.5
Coordinate loss weight	0.2

Table 21. Hyperparameter search space for combined neural network tuning

Hyperparameter	**Search Space**
Continent branch hidden dimensions	[128, 64] or [256, 128, 64]
City branch hidden dimensions	[128, 64] or [256, 128, 64]
Coordinate branch hidden dimensions	[128, 64] or [256, 128, 64]
Continent dropout initial	0.2 to 0.5
Continent dropout final	0.6 to 0.8
City dropout initial	0.2 to 0.5
City dropout final	0.6 to 0.8
Coordinate dropout initial	0.1 to 0.3
Coordinate dropout final	0.4 to 0.6
Learning rate	1e-4 to 1e-2 (log uniform)
Weight decay	1e-6 to 1e-3 (log uniform)
Batch normalization	True or False
Batch size	64, 128, 256
Continent loss weight	1.0 to 2.0
City loss weight	0.5 to continent_weight
Coordinate loss weight	0.05 to city_weight

6.3 GrowNet Parameters

Table 22. Default parameters for hierarchical GrowNet model

Parameter	**Value**
Architecture parameters	
Hidden size	256
Input feature dimension	200
Coordinate dimension	3
Dropout rates (2 layers)	0.2, 0.4
Boosting parameters	
Number of weak learners	30
Boost rate	0.4
Epochs per stage	20
Corrective epochs	5
Training parameters	
Learning rate	1e-3
Weight decay	1e-4
Batch size	128
Early stopping steps	5
Gradient clip	1.0
Loss weights	
Continent loss weight	2.0
City loss weight	1.0
Coordinate loss weight	0.5

Table 23. Hyperparameter search space for GrowNet tuning

Hyperparameter	Search Space
Hidden size	128, 256, 512
Number of weak learners	10 to 30
Boost rate	0.1 to 0.8
Learning rate	1e-4 to 1e-2 (log uniform)
Batch size	64, 128, 256
Weight decay	1e-6 to 1e-3 (log uniform)
Epochs per stage	5 to 10
Gradient clip	0.5 to 2.0
Hierarchical loss weights	
Continent loss weight	1.0 to 2.0
City loss weight	0.5 to (continent_weight - 0.05)
Coordinate loss weight	0.05 to (city_weight - 0.05)

6.4 Ensemble Meta-Model Parameters

6.4.1 XGBoost Parameters

Table 24. Default parameters for XGBoost models

Parameter	Classification	Regression
Objective	multi:softprob	reg:squarederror
Eval metric	mlogloss	rmse
Learning rate	0.1	0.1
Max depth	6	6
Min child weight	1	1
Gamma	0	0
Subsample	0.8	0.8
Colsample bytree	0.8	0.8
Lambda	1.0	1.0
Alpha	0.0	0.0
n_estimators	300	300

Table 25. Hyperparameter search space for XGBoost tuning

Hyperparameter	Search Space
Learning rate	1×10^{-3} to 0.3 (log uniform)
Max depth	3 to 12
Min child weight	1 to 10
Gamma	0 to 5
Subsample	0.5 to 1.0
Colsample bytree	0.5 to 1.0
Lambda	1×10^{-3} to 10 (log uniform)
Alpha	1×10^{-3} to 10 (log uniform)
n_estimators	100 to 400

6.4.2 LightGBM Parameters

Table 26. Default parameters for LightGBM models

Parameter	Classification	Regression
Objective	multiclass	regression
Metric	multi_logloss	rmse
Learning rate	0.1	0.1
Max depth	6	6
Num leaves	31	–
Min child samples	20	20
Subsample	0.8	0.8
Colsample bytree	0.8	0.8
Reg alpha	0.1	0.0
Reg lambda	1.0	1.0
n_estimators	300	300

Table 27. Hyperparameter search space for LightGBM tuning

Hyperparameter	Search Space
Learning rate	1×10^{-3} to 0.3 (log uniform)
Max depth	3 to 12
Num leaves	15 to 256 (classification only)
Min child samples	5 to 100
Subsample	0.5 to 1.0
Colsample bytree	0.5 to 1.0
Reg lambda	1×10^{-3} to 10 (log uniform)
Reg alpha	1×10^{-3} to 10 (log uniform)
n_estimators	100 to 400

6.4.3 CatBoost Parameters

Table 28. Default parameters for CatBoost models

Parameter	Classification	Regression
Loss function	MultiClass	RMSE
Eval metric	–	RMSE
Iterations	300	300
Learning rate	0.1	0.1
Depth	6	6
L2 leaf reg	3.0	3
Random strength	–	1
Bagging temperature	–	1
Border count	–	254
Random seed	42	42
Verbose	False	False

Table 29. Hyperparameter search space for CatBoost tuning

Hyperparameter	Search Space
Iterations	100 to 400 (classification), 100 to 500 (regression)
Learning rate	1×10^{-3} to 0.3 (log uniform)
Depth	3 to 10
L2 leaf reg	1 to 10
Random strength	1×10^{-9} to 10 (log uniform, regression only)
Bagging temperature	0 to 10 (regression only)
Border count	1 to 255 (regression only)

6.4.4 GrowNet Parameters

Table 30. Default parameters for GrowNet models (ensemble context)

Parameter	Classification	Regression
Hidden size	256	256
Num weak learners	10	10
Boost rate	0.4	0.4
Learning rate	1e-3	1e-3
Weight decay	1e-5	1e-5
Batch size	128	128
Epochs per stage	30	30
Early stopping steps	7	7
Gradient clip	1.0	1.0
n_outputs	–	3

Table 31. Hyperparameter search space for GrowNet tuning (ensemble context)

Hyperparameter	Search Space
Hidden size	128, 256, 512
Num weak learners	10 to 30
Boost rate	0.1 to 0.8
Learning rate	1×10^{-4} to 1×10^{-2} (log uniform)
Batch size	64, 128, 256
Weight decay	1×10^{-6} to 1×10^{-3} (log uniform)
Epochs per stage	5 to 10
Gradient clip	0.5 to 2.0

6.4.5 Neural Network (MLP) Parameters

Table 32. Default parameters for neural network (MLP) models (ensemble context)

Parameter	Classification	Regression
Input dimension	200	200
Hidden dimensions	[128, 64]	[128, 64]
Output dimension	7	3
Batch normalization	True	True
Initial dropout	0.3	0.2
Final dropout	0.8	0.5
Learning rate	1e-3	1e-3
Weight decay	1e-5	1e-5
Batch size	128	128
Epochs	400	400
Early stopping steps	20	50
Gradient clip	1.0	1.0

Table 33. Hyperparameter search space for neural network (MLP) tuning (ensemble context)

Hyperparameter	Search Space
Hidden dimensions	[64], [128], [128, 64], [256, 128, 64], [256, 128], [512, 256, 128, 64]
Initial dropout	0.1 to 0.3
Final dropout	0.5 to 0.8
Learning rate	1×10^{-4} to 1×10^{-2} (log uniform)
Batch size	64, 128, 256
Weight decay	1×10^{-6} to 1×10^{-3} (log uniform)
Gradient clip	0.5 to 2.0

6.4.6 TabPFN Parameters

Table 34. TabPFN model configuration

Parameter	Value
Model	Pre-trained TabPFN
Hyperparameter tuning	Max time

In-Radius Accuracy: Per-Continent

Table 35. Per-Continent In-Radius Accuracy (Separate Neural Network)

Continent	<1km	<5km	<50km	<100km	<250km	<500km	<1000km	<5000km
east_asia	0.0	0.0	0.0	0.0	0.0	0.0	0.00	28.42
europe	0.0	0.0	0.0	0.0	0.0	0.0	0.35	79.51
middle_east	0.0	0.0	0.0	0.0	0.0	0.0	0.00	93.33
north_america	0.0	0.0	0.0	0.0	0.0	4.7	9.40	88.59
oceania	0.0	0.0	0.0	0.0	0.0	0.0	0.00	0.00
south_america	0.0	0.0	0.0	0.0	0.0	0.0	0.00	4.76
sub_saharan_africa	0.0	0.0	0.0	0.0	0.0	0.0	0.00	3.39

Table 36. Per-Continent In-Radius Accuracy (Combined Neural Network)

Continent	<1km	<5km	<50km	<100km	<250km	<500km	<1000km	<5000km
east_asia	0.0	0.0	0.0	0.0	0.0	0.0	0.00	97.84
europe	0.0	0.0	0.0	0.0	0.0	0.0	0.35	95.76
middle_east	0.0	0.0	0.0	0.0	0.0	0.0	0.00	100.00
north_america	0.0	0.0	0.0	0.0	0.0	4.7	9.40	97.99
oceania	0.0	0.0	0.0	0.0	0.0	0.0	0.00	55.56
south_america	0.0	0.0	0.0	0.0	0.0	0.0	0.00	85.71
sub_saharan_africa	0.0	0.0	0.0	0.0	0.0	0.0	0.00	100.00

Table 37. Per-Continent In-Radius Accuracy (GrowNet)

Continent	<1km	<5km	<50km	<100km	<250km	<500km	<1000km	<5000km
east_asia	0.0	0.0	2.52	6.83	25.18	50.00	69.06	96.40
europe	0.0	0.0	0.00	0.00	4.24	19.79	60.07	86.22
middle_east	0.0	0.0	0.00	0.00	0.00	26.67	66.67	93.33
north_america	0.0	0.0	0.00	0.67	4.03	16.11	32.21	87.25
oceania	0.0	0.0	0.00	0.00	0.00	0.00	0.00	11.11
south_america	0.0	0.0	0.00	0.00	0.00	0.00	33.33	61.90
sub_saharan_africa	0.0	0.0	1.69	3.39	27.12	49.15	67.80	93.22

Table 38. Per-Continent In-Radius Accuracy (Ensemble, TabPFN)

Continent	<1km	<5km	<50km	<100km	<250km	<500km	<1000km	<5000km
east_asia	0.0	6.83	72.30	76.26	82.01	84.17	89.21	97.84
europe	0.0	6.01	66.08	69.96	77.74	82.69	86.57	95.76
middle_east	0.0	0.00	80.00	80.00	86.67	86.67	93.33	100.00
north_america	0.0	0.67	62.42	68.46	76.51	81.88	86.58	97.99
oceania	0.0	0.00	0.00	0.00	0.00	0.00	0.00	55.56
south_america	0.0	0.00	28.57	38.10	42.86	61.90	66.67	85.71
sub_saharan_africa	0.0	1.69	84.75	88.14	88.14	89.83	94.92	100.00