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

Accurate estimation of geographic origin of environmental samples from microbial signatures has important applications in biosurveillance, forensic science, and public health. The state-of-the-art tool at the time, mGPS, utilized a hierarchical XGBoost-based method to predict locations from microorganism sequence relative abundances. However, mGPS suffered some restrictions: (1) relatively poor coordinate prediction precision, (2) error propagation throughout the hierarchical prediction framework, and (3) a breakdown of scalability or extensibility to larger, more complex datasets.

To mitigate these issues, we evaluated a diverse range of models-such as neural networks, GrowNet, and advanced ensemble methods—on the MetaSUB dataset (4,070 samples from 40 cities across 7 continents). While several approaches were explored, our ensemble learning strategy, which combined XGBoost, CatBoost, LightGBM, TabPFN, neural networks, and GrowNet within hierarchical meta-models, delivered the most significant improvements. This approach achieved a tenfold reduction in median coordinate error (from 137 km with mGPS to 13.7 km), with modest gains in continent and city classification accuracy. Additionally, we introduced an error calculation framework that quantifies how misclassifications at broader levels propagate cascading errors to coordinate predictions, providing deeper insight into model performance.

These results demonstrate that ensemble learning, leveraging the complementary strengths of diverse models, is needed for accurate geographic prediction from highly variable biological data. Our optimized framework provides a new benchmark for spatial prediction from metagenomic profiles and provides a scalable platform for future public health, forensic science, and ecological applications. Better feature selection, modeling species interactions, and incorporation of autoencoder-based representations will be the focus of future research to further enhance predictive accuracy and robustness.

1. Introduction

1.1 Geographical Prediction Using Microbial Signatures

Microorganisms from environmental samples harbor biological signatures of local environmental conditions, human activity, and ecological processes from specific regions (Zhang et al., 2024). This property enables uses in biosurveillance, forensics, and public health monitoring (Robinson et al., 2021).

The microbial Global Population Structure (mGPS) tool takes advantage of these signatures for geographical prediction using relative sequence abundance (RSA) analysis of microorganisms (Zhang et al., 2024). The original implementation used a hierarchical XGBoost model for continent, city, and coordinate prediction, with 92% high city-level accuracy and low 137km median error distance on the MetaSUB dataset (Zhang et al., 2024).

1.2 Previous Work and Methodology

Building on the mGPS framework, most studies employ XGBoost with improvements such as hyperparameter optimization and recursive feature elimination (RFE) to reduce thousands of microbial features to a more informative subset. (Bergman, 2025) The typical workflow is hierarchical: continent $\rightarrow$ city $\rightarrow$ coordinates, with prediction probabilities at each level used to inform subsequent predictions. (Zhang et al., 2024)

1.3 Limitations in Existing Approaches

Current approaches have several clear limitations. First, the hierarchical structure of prediction (continent $\rightarrow$ city $\rightarrow$ coordinates) means that errors at higher levels, such as misclassifying the continent or city, directly propagate and can result in large errors in the final coordinate predictions. This cascading effect can significantly degrade overall model accuracy (Liu et al., 2025). Second, previous methodologies often report coordinate prediction accuracy based on the assumption that continent and city have been correctly classified, but do not clearly specify this dependency. This can make the reported metrics misleading, as high accuracy at one level may mask errors at subsequent levels, and the evaluation criteria for hierarchical prediction are not always well defined or transparent (Kosmopoulos et al., 2014). Third, most of the current approaches rely on XGBoost, which is highly effective for small to medium-sized tabular datasets (typically up to several thousand samples). However, as larger and more diverse datasets become available, these methods may not scale well or fully leverage the available data. More sophisticated approaches, such as deep learning models, may be required to handle larger datasets and capture complex patterns, but this limitation has not been adequately addressed in prior work (Tang, 2024).

1.4 Research Objectives and Contributions

This study pursues several key objectives in hierarchical geographic prediction of microbial samples. One major aim is to minimize error propagation that can occur with hierarchical predictions, since mistakes made at higher levels (continent or city) are likely to produce substantial errors in the final coordinates. We also introduce a new mathematical framework to explicitly describe hierarchical errors, providing a rigorous and transparent understanding of how errors propagate in the prediction hierarchy. Additionally, this work develops a model capable of incorporating larger and more diverse datasets than previous studies, improving both the scale and accuracy of geographic prediction from microbial samples.

1.5 Dataset and Proposed Improvements

Before quality control, the global atlas contained a total of 4,728 metagenomic samples collected from mass-transit systems in 60 cities spanning a three-year period (Danko et al., 2021). Following a basic quality control, the seven cities with unclear geographical coordinates were removed, leaving in 4,135 samples from 53 cities for biogeographical analysis (Danko et al., 2021). Further post-quality control filtering involved removing cities which had fewer than eight samples, leaving 4,070 samples from 40 cities. This post-QC dataset was used to construct the mGPS model (Zhang et al., 2024). The dataset is geographically diverse, with sample counts varying widely between cities and continents (Figure 1). For example, Europe and Asia-Pacific are strongly represented, whereas Oceania and sub-Saharan Africa are poorly represented. Similarly, some cities such as New York City, Hong Kong, and London are strongly represented, whereas cities like Brisbane, Auckland, and São Paulo are very poorly represented (Danko et al., 2021).

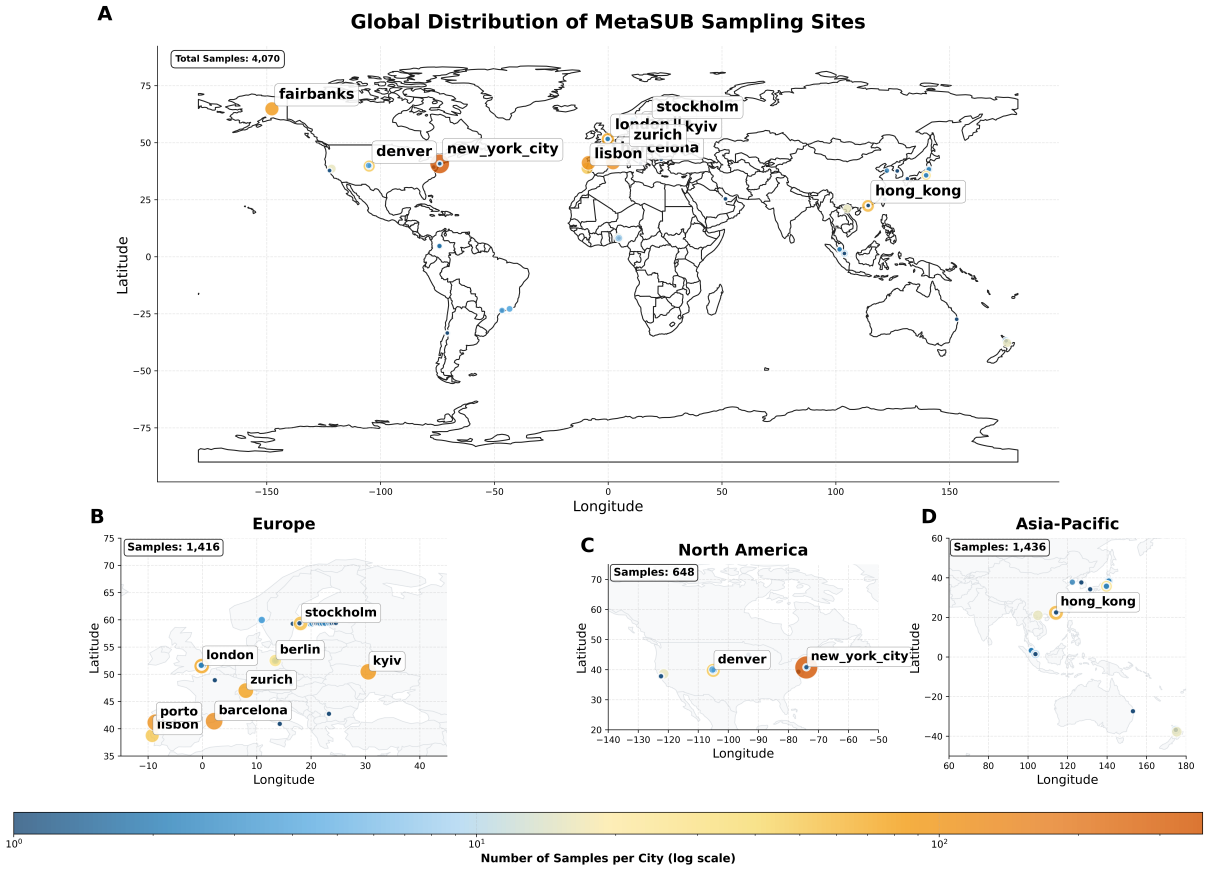

Figure 1. *Global distribution of MetaSUB sampling sites. (A) World map showing sample locations and counts. (B-D) Regional breakdowns for Europe, North America, and Asia-Pacific. The color scale indicates the number of samples per city (log scale).*

Each sample contains a taxonomic profile with relative sequence abundances, reduced to 200-300 informative features from 3000 via RFE (Zhang et al., 2024). The taxonomic diversity is dominated by bacteria, with minor representation from eukaryotes, viruses, and archaea (Figure 2). At finer taxonomic levels, the dataset is rich in Pseudomonadota, Actinomycetota, and Bacillota, among others.

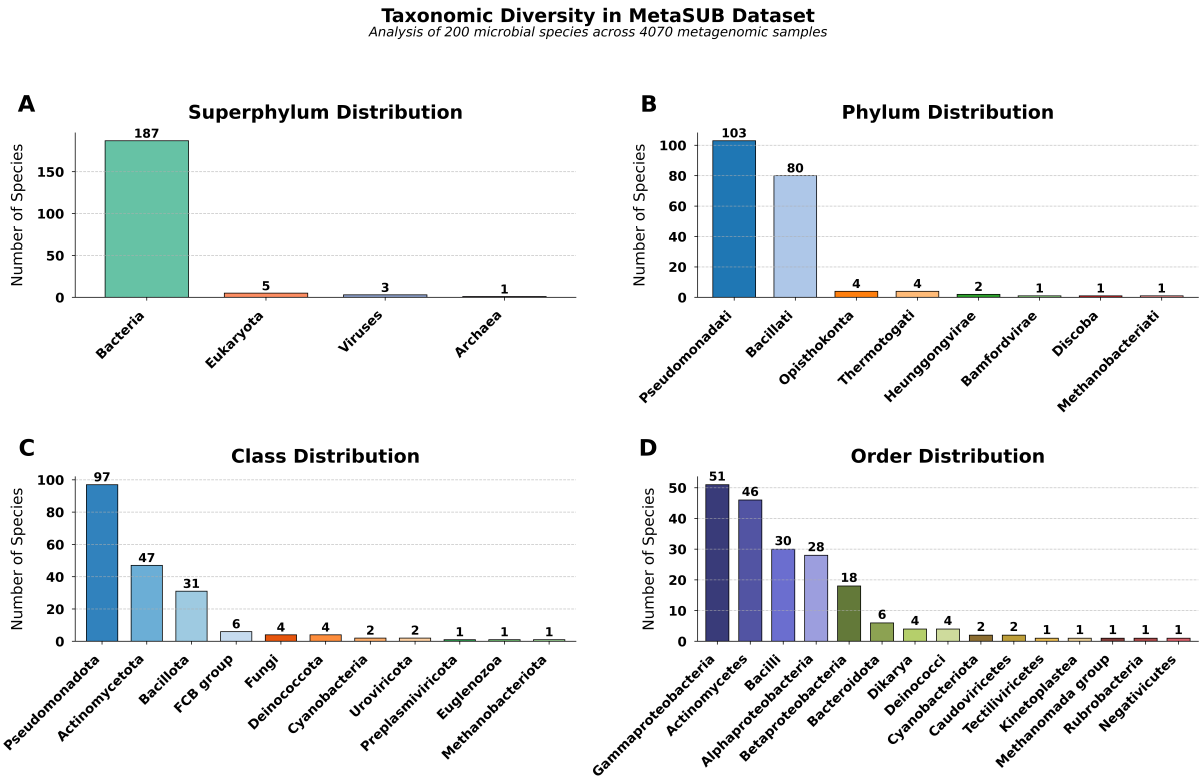

Figure 2. *Taxonomic diversity in the MetaSUB dataset. (A) Superphylum, (B) Phylum, (C) Class, and (D) Order distributions for 200 microbial species. The dataset originally included 4,728 samples before quality control, and 4,070 post-QC samples from 40 cities were used for analysis. Bacteria dominate the dataset, with Pseudomonadota and Actinomycetota as major groups.*

2. Materials and Methods

2.1 Dataset and Preprocessing

We analyzed the MetaSUB dataset from the original mGPS study (Zhang et al., 2024), accessed via their GitHub repository. This dataset comprises 4,070 quality-controlled samples collected from subway stations in 40 cities across 7 continents between 2016 and 2017. Each sample contains taxonomic profiles with relative sequence abundances, generated by subsampling to 100,000 classified reads and processed using KrakenUniq with the NCBI/RefSeq Microbial database (Danko et al., 2021).

To maintain methodological consistency with previous mGPS work, we applied the same quality control and feature selection procedures. Specifically, cities with fewer than eight samples were excluded, and recursive feature elimination (RFE) with Random Forest was used to reduce the initial set of approximately 3,000 microbial features to the 200–300 most informative, using 5-fold cross-validation (Guyon et al., 2002). Class imbalance—particularly for underrepresented continents such as Oceania and Africa—was addressed using the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002), achieving a 1:3 ratio between minority and majority classes. These steps ensured that our dataset and preprocessing pipeline remained directly comparable to the original mGPS study.

2.2 Model Development

We developed several modeling approaches to address the hierarchical geographic prediction problem, each offering distinct advantages and characteristics.

2.2.1 Neural Networks

Neural networks were chosen as a core modeling approach due to their capacity to learn complex, non-linear relationships and, crucially, their scalability with increasing data size (LeCun et al., 2015). The primary motivation was to develop a robust model that could not only perform well on the current dataset but also generalize effectively as more data becomes available in the future. This makes neural networks particularly suitable for scenarios where data volume is expected to grow, ensuring the modeling framework remains adaptable and performant.

Separate Neural Network Models In accordance with the previous study, which utilized a hierarchical approach with XGBoost (Zhang et al., 2024)(Chen and Guestrin, 2016), we constructed a set of independent neural networks to serve as baselines and to analyze error propagation at each prediction level. Specifically, we developed 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 4 and 5.

Each network architecture follows a progressive dropout, a batch normalization, and ReLU activation functions.

Coordinate Transformation for Geographical Prediction: To appropriately model the spherical geometry of the Earth and avoid issues such as gradient explosion, vanishing gradients, and improper scaling, we transform latitude (ϕ) and longitude (λ) into 3D Cartesian coordinates for all neural network-based coordinate prediction models (Snyder, 1987; Aydin et al., 2016). This transformation ensures that points close on the globe (e.g., near the $-180°/+180°$ longitude boundary) are also close in the transformed space, which is not the case if standard scaling is applied directly to latitude and longitude. The transformation is defined as:

$$\begin{aligned} x &= \cos(\phi)\cos(\lambda) \\ y &= \cos(\phi)\sin(\lambda) \\ z &= \sin(\phi) \end{aligned} \tag{1}$$

For evaluation, we apply the inverse transformation to the predicted (x, y, z) values, converting them back to latitude and longitude in radians, and then to degrees. This allows for accurate geodesic error computation and ensures that the model predictions are interpretable in the original coordinate system.

Each neural network in the separate hierarchy is trained independently using a standard loss function appropriate for its task. For continent and city classification, cross-entropy loss is employed (Paszke et al., 2019), with optional class weights to address class imbalance:

$$\mathcal{L}_{\text{classification}} = \text{CrossEntropyLoss}(\text{predictions}, \text{targets}, \text{weight} = w_{\text{class}}) \tag{2}$$

For coordinate regression, mean squared error (MSE) (Paszke et al., 2019) loss is used:

$$\mathcal{L}_{\text{regression}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2 \tag{3}$$

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

Supplementary Table 3 provides a description of the architecture and training settings used for each separate neural network.

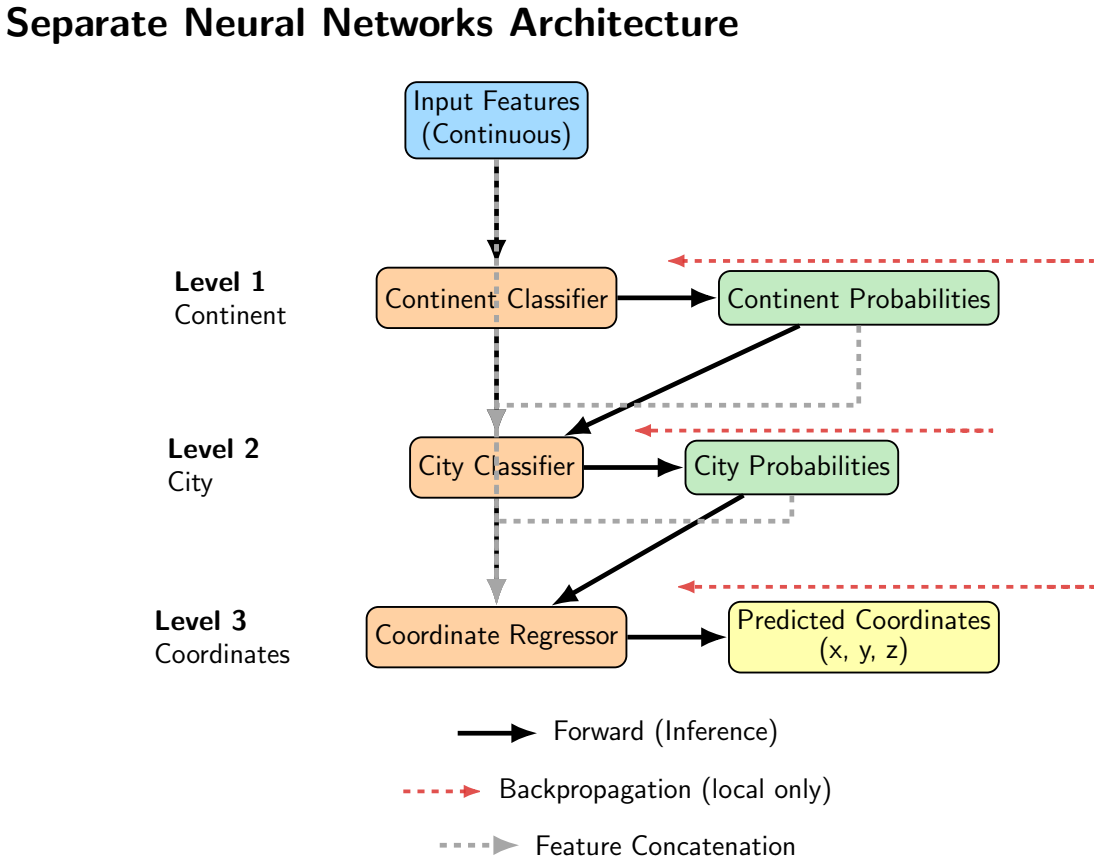

Figure 3. *Schematic of the separate neural network approach for hierarchical geographic prediction. Each prediction level (continent, city, coordinates) is modeled by an independent neural network. Outputs from each level are used as inputs for the next, but training and backpropagation are performed 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 Neural Networks To enable end-to-end hierarchical learning, we developed the Combined Neural Networks, 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. As with the separate models, coordinate prediction in this architecture also employs the Cartesian transformation described in Equation 1 (Snyder, 1987; Aydin et al., 2016).

Default parameters and the hyperparameter search space for the Combined Neural Networks are provided in Supplementary Tables 7 and 8.

The total weighted loss for the combined neural network is defined as:

$$\mathcal{L}_{\text{total}} = w_1\mathcal{L}_{\text{continent}} + w_2\mathcal{L}_{\text{city}} + w_3\mathcal{L}_{\text{coordinate}} \tag{4}$$

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 by penalizing errors at higher levels more strongly, reflecting the hierarchical structure of the problem. During backpropagation, gradients flow through all branches, but their magnitudes are modulated by these weights, promoting robust feature learning across the hierarchy.

The architecture and training parameters for the Combined Neural Networks are summarized in Supplementary Table 6.

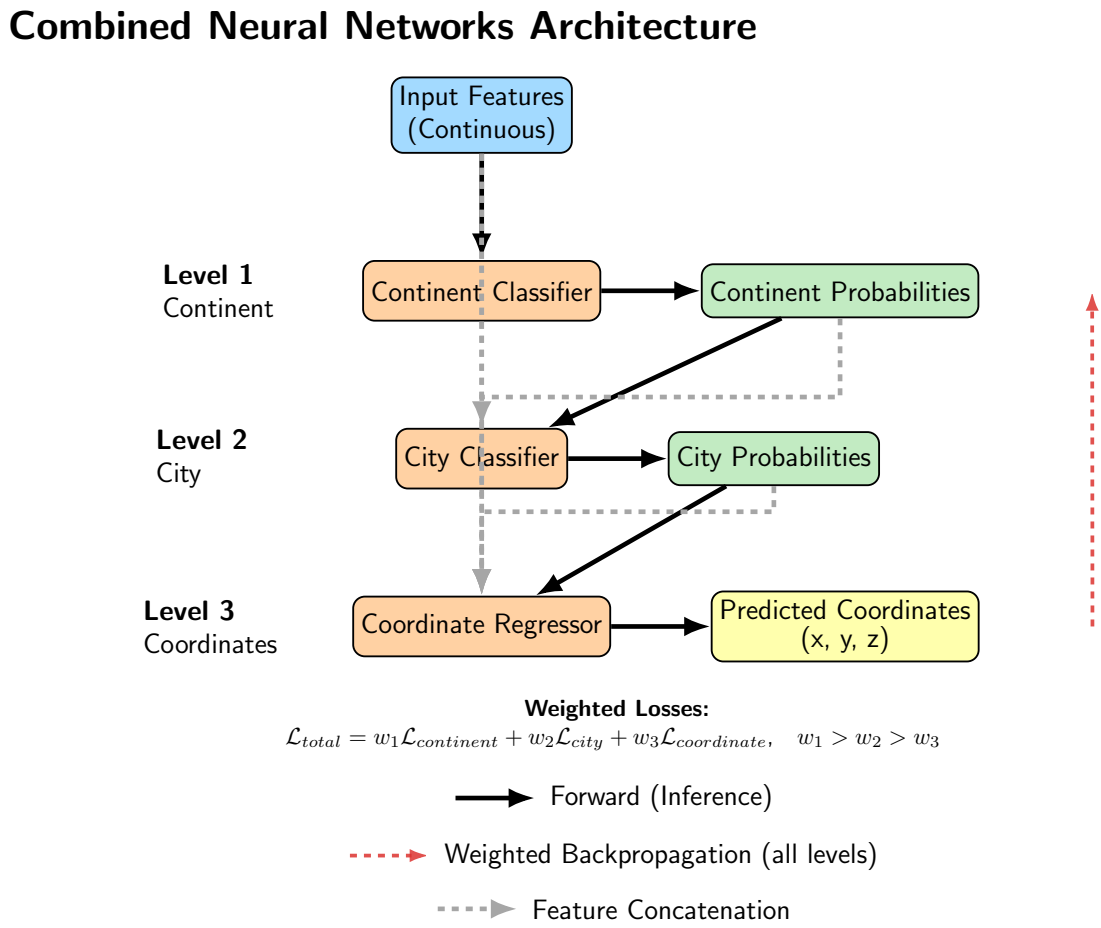

Figure 4. *Diagram of the Combined Neural Networks architecture. This unified multi-task neural network consists of sequential branches for continent, city, and coordinate prediction. Feature representations are shared, and predictions from higher levels are concatenated with features for downstream tasks. Training uses a weighted multi-task loss to reflect the hierarchy. Backpropagation occurs through all branches, allowing the model to learn robust representations that minimize error propagation.*

In the original (separate neural network) approach, each model is trained independently and the loss is propagated only within that level of the hierarchy. This limits the ability of the models to learn shared representations and can lead to error propagation across levels. In contrast, the combined neural network architecture enables end-to-end hierarchical learning, where the loss function is propagated through the entire hierarchy. Joint optimization allows gradients to flow through all levels, encouraging the model to

learn feature representations that minimize errors both locally and throughout the hierarchy. As a result, the combined neural network approach is better equipped to handle task-level dependencies and reduce compounding errors, which led to improved overall performance (Ruder, 2017).

2.2.2 GrowNet Architecture

We sought a model that could leverage the boosting principle—proven highly effective in tabular data settings by algorithms such as XGBoost (Chen and Guestrin, 2016)—while also benefiting from the flexibility and scalability of neural networks, which are known to perform better as dataset size increases (Tang, 2024). GrowNet (Feng et al., 2021) was chosen because it closely follows the boosting approach of XGBoost, but replaces decision trees with neural networks as weak learners (Feng et al., 2021).

GrowNet is a gradient boosting framework that employs neural networks as weak learners for multi-task learning (Feng et al., 2021). The algorithm proceeds by sequentially adding shallow neural networks to the ensemble, each trained to correct the residuals (pseudo-residuals) of the previous learners, analogous to boosting in XGBoost (Chen and Guestrin, 2016). At each stage m, the pseudo-residuals $\mathbf{r}^{(m)}$ are computed as the negative gradient of the loss with respect to the current ensemble prediction, i.e., $\mathbf{r}^{(m)} = -\nabla_{F^{(m-1)}} \mathcal{L}(y, F^{(m-1)})$. Each weak learner h_m is then trained to fit these residuals.

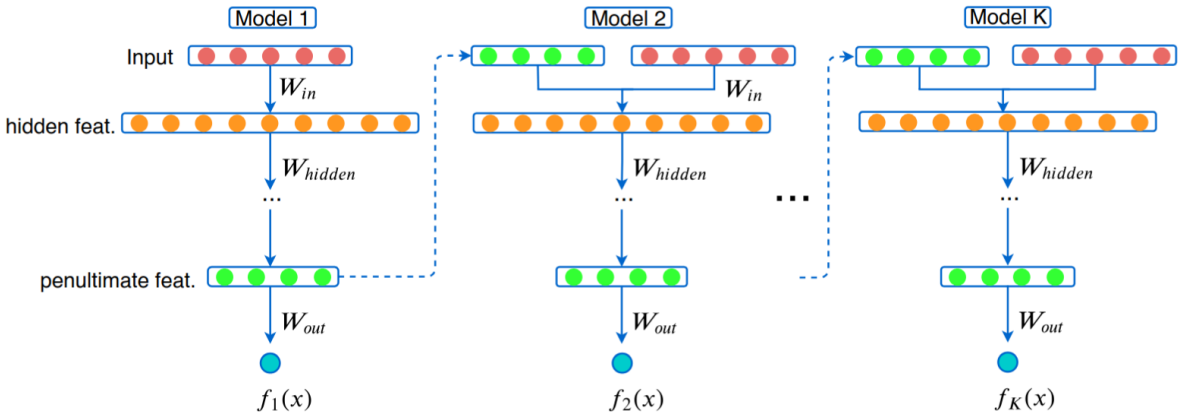

Figure 5. *Diagram of the GrowNet architecture. This framework utilizes a multi-task learning approach with neural networks as weak learners, enabling effective handling of hierarchical tasks. Figure obtained from (Feng et al., 2021).*
Notation correspondence: *In this work, the ensemble prediction at stage m is denoted $F^{(m)}$, the pseudo-residuals as $\mathbf{r}^{(m)}$, and each weak learner as h_m. These may be labeled differently in the original figure, but the underlying concepts are equivalent.*

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 $\mathbf{r}^{(m)} = -\nabla_{F^{(m-1)}} \mathcal{L}(y, F^{(m-1)})$.

   (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:

$$\{\theta_1, \ldots, \theta_m\} \leftarrow \arg\min_{\{\theta_i\}} \mathcal{L}_{\text{total}}(F^{(m)}; \{\theta_i\}_{i=1}^m) \tag{5}$$

   (f) Evaluate on validation data and apply early stopping if necessary.

4. Return the final ensemble $\mathcal{F} = \{h_1, \ldots, h_M\}$.

Here, $F^{(m)}$ is the current ensemble prediction, h_m is the m-th weak learner, ρ is the learning rate, and $\mathcal{L}_{\text{total}}$ is the composite loss function (see Equation 4). Pseudo-residuals represent the direction and magnitude by which the current model's predictions should be adjusted to minimize the loss. The corrective optimization step enables earlier weak learners to adapt based on information acquired by subsequent learners, enhancing ensemble coherence and predictive performance.

In simple terms, GrowNet builds an ensemble of neural networks, each one learning to correct the mistakes of the previous ones. At each stage, the model computes how much its current prediction is wrong (the pseudo-residual), fits a new neural network to these errors, and adds it to the ensemble. This process continues for several stages, and occasionally all networks are jointly fine-tuned to further reduce the overall error. This approach allows GrowNet to combine the flexibility of neural networks with the boosting principle, resulting in strong performance for hierarchical, multi-task problems.

2.2.3 Ensemble Learning

Model Selection and Integration Strategy: Our ensemble strategically combines different models to minimize hierarchical error across varying data regimes: gradient boosting models (XGBoost, LightGBM, CatBoost), TabPFN, neural networks (MLPs), and GrowNet. This selection balances proven effectiveness on tabular data with scalability for larger datasets, ensuring robust performance across different data scenarios (Chen and Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018; Grinsztajn et al., 2022; Hütter

et al., 2022; Feng et al., 2021; Caruana et al., 2008; Tang, 2024; Dietterich, 2000; Opitz and Maclin, 1999). The ensemble employs task-specific integration mechanisms: classification tasks use threshold filtering with XGBoost meta-models (Supplementary Table 22) to leverage diverse model strengths, while regression tasks select only the best-performing single model to preserve granular predictions, as illustrated in Figure 6.

Model Architecture and Hyperparameters: The ensemble incorporates the following models. Gradient boosting models—including XGBoost (Chen and Guestrin, 2016) (see Supplementary Tables 11, 12), LightGBM (Ke et al., 2017) (Supplementary Tables 13, 14), and CatBoost (Prokhorenkova et al., 2018) (Supplementary Tables 15, 16)—are optimized for capturing non-linear relationships in tabular data. TabPFN (Hütter et al., 2022) is a prior-data fitted neural network leveraging meta-learning for rapid adaptation to new tabular tasks (see Supplementary Table 21). Standard multilayer perceptrons provide capacity for complex feature interactions at scale (Supplementary Tables 19, 20). GrowNet (Feng et al., 2021) is a gradient boosting neural network architecture offering robust performance for larger datasets with intricate relationships (Supplementary Tables 17, 18).

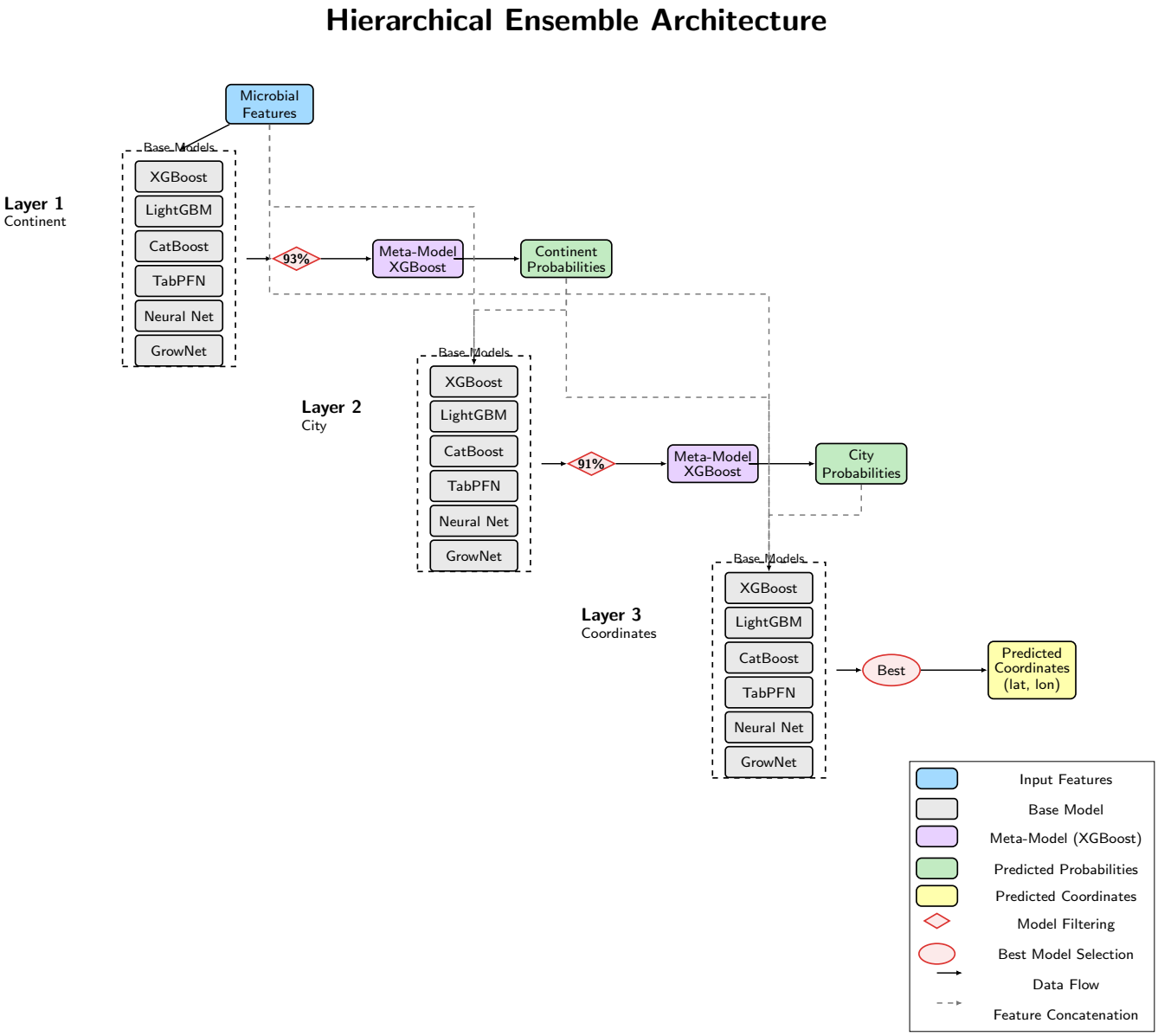

Figure 6. *Overview of the hierarchical ensemble learning workflow. The ensemble is organized in three layers: continent classification, city classification, and coordinate regression. At each stage, predictions from multiple base models are combined using meta-models, and probability outputs are used as augmented features for subsequent layers.*

Hierarchical Ensemble Implementation The hierarchical ensemble architecture, as illustrated in Figure 6, consists of three layers, each with a distinct strategy tailored to the prediction task. In Layer 1 (Continent Classification), multiple base models predict continent probabilities from microbial features, with SMOTE applied to address class imbalance. Only models exceeding a 93% accuracy threshold are retained. Each retained model could be independently optimized using Bayesian optimization (Optuna (Akiba et al., 2019)) to further enhance performance. TabPFN, however, does not undergo conventional hyperparameter tuning and instead uses a fixed training time parameter (`max_training_time`), which controls how long the model explores its internal pre-defined

configurations to find the optimal fit for the data (maximum of 300 seconds). For each selected and tuned model, out-of-fold (OOF) predictions are generated using 5-fold cross-validation: in each fold, the model is trained on $k-1$ folds (with tuned hyperparameters) and predicts on the held-out fold. This ensures that every OOF prediction is made by a model that has not seen the corresponding sample during either training or hyperparameter selection, thus preventing information leakage. The concatenated OOF predictions from all selected models are used as meta-features to train the meta-model (e.g., XGBoost), which learns to optimally combine the base models' outputs. For TabPFN models that pass the threshold, we always use the maximum training time setting for both OOF prediction generation and final model training to ensure optimal adaptation to the dataset. Layer 2 (City Classification) builds on this by using both the original microbial features and continent probabilities from Layer 1; models surpassing a 91% accuracy threshold are included, and the same meta-learning protocol is followed to leverage the diverse inductive biases of different models, as some excel at predicting specific geographic regions. Layer 3 (Coordinate Prediction) utilizes the complete feature set—microbial abundances, continent probabilities, and city probabilities—but, unlike the classification layers, selects only the single best-performing model for final predictions. This is because averaging continuous regression outputs can degrade performance by smoothing strong individual predictions (Dietterich, 2000; Opitz and Maclin, 1999). For coordinate regression, two approaches are evaluated: tree-based models predict latitude first, followed by longitude conditioned on the predicted latitude, while neural networks directly predict 3D Cartesian coordinates (see Equation 1 (Snyder, 1987; Aydin et al., 2016)), which are subsequently converted to latitude and longitude. The model achieving the lowest median Haversine distance error is selected for final predictions. This dynamic selection mechanism, depicted in Figure 6, allows the ensemble to adapt as datasets grow, transitioning from tree-based models to neural networks when data volume increases (Tang, 2024).

2.3 Geodesic Error Calculation and Error Propagation

Geodesic Error Calculation (Haversine Formula) Geodesic error is computed as the great-circle distance between predicted and true coordinates using the Haversine formula:

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \tag{6}$$

where:

- d is the geodesic distance (in kilometers),
- R is the Earth's radius (mean value $R = 6371$ km),
- ϕ_1, ϕ_2 are the latitudes (in radians) of the true and predicted points,

- λ_1, λ_2 are the longitudes (in radians) of the true and predicted points,
- $\Delta\phi = \phi_2 - \phi_1$,
- $\Delta\lambda = \lambda_2 - \lambda_1$.

This formula accurately measures the shortest distance over the Earth's surface between two points, and is used throughout this work to quantify spatial prediction error.

To provide a more nuanced understanding of coordinate prediction error, we compute the expected coordinate error $E[D]$ as a weighted sum over all possible combinations of continent and city prediction correctness:

$$E(D) = P_{cc,zc}\, E_{cc,zc} + P_{cc,zi}\, E_{cc,zi} + P_{ci,zc}\, E_{ci,zc} + P_{ci,zi}\, E_{ci,zi} \tag{7}$$

where:

- $P_{cc,zc} = P(C = C^*, Z = Z^*)$ is the probability of predicting both the correct continent and correct city,
- $P_{cc,zi} = P(C = C^*, Z \neq Z^*)$ is the probability of predicting the correct continent but incorrect city,
- $P_{ci,zc} = P(C \neq C^*, Z = Z^*)$ is the probability of predicting the incorrect continent but correct city,
- $P_{ci,zi} = P(C \neq C^*, Z \neq Z^*)$ is the probability of predicting both the incorrect continent and incorrect city,
- $E_{cc,zc} = E(D|C = C^*, Z = Z^*)$ is the expected geodesic error when both continent and city are correct,
- $E_{cc,zi} = E(D|C = C^*, Z \neq Z^*)$ is the expected error when continent is correct but city is incorrect,
- $E_{ci,zc} = E(D|C \neq C^*, Z = Z^*)$ is the expected error when continent is incorrect but city is correct,
- $E_{ci,zi} = E(D|C \neq C^*, Z \neq Z^*)$ is the expected error when both continent and city are incorrect.

This decomposition quantifies how errors at the continent and city levels propagate to the final coordinate prediction.

3. Results

3.1 Overview

This section presents the performance evaluation of various hierarchical machine learning models for geographic prediction using m etagenomic data. We compare the effectiveness of separate neural networks, combined neural networks, GrowNet, and ensemble learning approaches in predicting geographic origins at continent, city, and coordinate levels.

3.2 Dataset and Evaluation Metrics

We evaluated all models on the filtered MetaSUB dataset, containing 4,070 samples from 40 cities on 7 continents. Data were partitioned into training, validation, and test sets (2,604/652/814 samples, respectively) after quality control. The dataset exhibits class imbalance, particularly at the continent and city levels.

Principal metrics of evaluation are **classification accuracy**, **macro-averaged F1-score**, and **weighted F1-score** for categorical predictions at both continent and city scales. For geospatial accuracy estimation, we measured **geodesic error**, the great-circle distance between predicted and actual coordinates on Earth's surface. 6 We also provide **in-radius accuracy** (the proportion of predictions within specified geodesic distances of the true location). On classification tasks, **AUPR** (area under the precision-recall curve) and **AUC** (area under the ROC curve) are only reported for the ensemble model to facilitate a balanced comparison with the mGPS state-of-the-art model. (Zhang et al., 2024)

3.3 Evaluation Metrics Explanation

Evaluation Metrics Defined

Accuracy: Proportion of correct predictions among all samples.

Macro-averaged F1-score: F1-score computed independently for each class and then averaged, treating all classes equally.

Weighted F1-score: F1-score computed for each class and averaged using the number of true instances per class as weights; more robust to class imbalance.

Geodesic error: Great-circle distance (in km) between predicted and true coordinates on Earth's surface.

In-radius accuracy: Proportion of predictions within a specified geodesic distance (e.g., 50 km, 100 km, etc.) from the true location.

R^2 (Coefficient of determination): Proportion of variation in the true coordinates explained by the model.

AUC (Area Under the ROC Curve): Measures the ability of the model to distinguish between classes, summarizing the trade-off between true positive rate and false positive rate across thresholds.

AUPR (Area Under the Precision-Recall Curve): Evaluates the trade-off between precision and recall, especially informative for imbalanced datasets.

3.4 Model Performance

This section presents the performance of the various models evaluated on the MetaSUB dataset, focusing on continent and city classification accuracy, geodesic error, and in-radius accuracy. The results are summarized in Table 1.

Table 1. Comparison of model performance across continent and city metrics, and error group analysis.

Model	Continent Metrics			City Metrics			Cc-Zc				Cc-Zi				Ci-Zc				Ci-Zi			
	Acc.	Avg F1	Wtd F1	Acc.	Avg F1	Wtd F1	Mean	Median	Prop.	Wtd	Mean	Median	Prop.	Wtd	Mean	Median	Prop.	Wtd	Mean	Median	Prop.	Wtd
Separate NN	0.85	0.78	0.85	0.70	0.55	0.71	3994	3255	0.694	2772	5333	3703	0.155	826	7668	8555	0.007	57	9098	7532	0.144	1308
Combined NN	0.83	0.75	0.83	0.75	0.45	0.72	502	274	0.714	358	2101	1523	0.113	237	3434	2252	0.036	122	6637	5377	0.138	913
GrowNet	0.86	0.77	0.86	0.75	0.60	0.76	904	599	0.742	671	2215	1710	0.122	269	4501	4324	0.009	39	7090	5896	0.128	906
Ensemble	0.95	0.89	0.95	0.93	0.80	0.92	208.1	12.3	0.903	187.9	2148.1	1713.5	0.045	97.6	3902.2	3534.2	0.022	86.3	7365.5	6822.9	0.029	217.2

Notes: Acc. = Accuracy; Avg F1 = Macro-averaged F1 score; Wtd F1 = Weighted F1 score. Error group columns: **Cc-Zc** = Continent correct, City correct; **Cc-Zi** = Continent correct, City incorrect; **Ci-Zc** = Continent incorrect, City correct; **Ci-Zi** = Continent incorrect, City incorrect. For each group: Mean/Median Error (km), Proportion of samples, and Weighted Error.

3.4.1 Separate Neural Networks

The separate neural network approach was evaluated in three sequential stages: continent classification, city classification, and coordinate regression.

Continent Classification The continent classifier achieved a test accuracy of 84.9% with a macro-averaged F1-score of 0.78 and a weighted F1-score of 0.85, indicating decent performance across continents despite class imbalance. Supplementary Table 24 presents detailed classification metrics.

City Classification The city classifier achieved a test accuracy of 70.1%, a macro-averaged F1-score of 0.55, and a weighted F1-score of 0.71. The lower macro-averaged F1-score compared to weighted F1-score reflects the effect of class imbalance, with underrepresented cities showing lower classification performance. Supplementary Table 28 presents a detailed city classification metrics.

Coordinate Regression The coordinate regression model achieved an coefficient of determination (R^2) of 0.658 on the test set before inverse transformation of predicted coordinates to latitude and longitude and a R^2 of -2.3622 after inverse transformation. Geodesic error analysis revealed a median error of 4,237 km, mean error of 4,962 km, and maximum error of 17,788 km. Supplementary Table 32 presents a detailed error breakdown by prediction correctness.

In-radius accuracy analysis revealed that only 1.8% of predictions were within 1,000 km of the true location, while 55.7% were within 5,000 km (Supplementary Table 36). These metrics indicate that the separate neural networks approach, while providing reasonable classification performance, struggles with precise coordinate prediction.

3.4.2 Combined Neural Networks

The combined hierarchical neural network jointly predicts continent, city, and coordinates using a unified architecture with weighted multi-task learning. On the test set, this model achieved 82.7% continent a ccuracy (macro F1-score: 0.75, weighted F1-score: 0.83; Supplementary Table 25) and 74.9% city accuracy (macro F1-score: 0.45, weighted F1-score: 0.72; Supplementary Table 29). For coordinate regression, the model achieved a R^2 of 0.7 before inverse transformation of predicted coordinates to latitude and longitude and an R^2 of 0.62 after inverse transformation. The median geodesic error decreased substantially to 519 km, with a mean error of 1,631 km and maximum error of 19,604 km. Supplementary Table 33 provides a detailed error analysis by prediction group. In-radius accuracy showed marked improvement, with 66.3% of predictions within 1,000 km and 89.3% within 5,000 km (Supplementary Table 36). These results demonstrate that the combined neural network approach significantly outperforms separate networks for coordinate prediction while maintaining comparable classification performance.

3.4.3 Hierarchical GrowNet

GrowNet, which combines neural networks with gradient boosting principles (Feng et al., 2021), achieved the highest classification accuracy among neural models. It reached 86.4% continent accuracy (macro F1-score: 0.77, weighted F1-score: 0.86; Supplementary Table 26) and 75.1% city accuracy (macro F1-score: 0.60, weighted F1-score: 0.76; Supplementary Table 30).

For coordinate regression, GrowNet achieved a median geodesic error of 823 km and mean error of 1,885 km, with a maximum error of 18,964 km. The coordinate regression R^2 was 0.685 and 0.627 before and after inverse transformation of predicted coordinates to latitude and longitude respectively. The in-radius accuracy was 57.4% within 1,000 km and 89.1% within 5,000 km (Supplementary Table 36). Supplementary Table 34 provides a detailed error analysis by prediction group. Compared to both separate and combined neural networks, GrowNet showed lesser performance in city prediction accuracy to the combined neural network approach.

3.4.4 Ensemble Learning Model

Our ensemble learning approach, which integrates multiple models , achieved state-of-the-art results across all prediction tasks. This superior performance aligns with empirical findings that ensemble methods often outperform individual models (Opitz and Maclin, 1999; Mahdavi-Shahri et al., 2016). The ensemble attained 95.0% continent accuracy (macro F1-score: 0.89, weighted F1-score: 0.95; Supplementary Table 27) and 93.0% city accuracy (macro F1-score: 0.80, weighted F1-score: 0.92; Supplementary Table 31), with TabPFN delivering exceptional coordinate regression performance.

Continent Classification The ensemble model achieved the highest continent classification accuracy (95.0%) among all approaches. Even for underrepresented continents like Oceania, the model maintained reasonable performance, with a macro-averaged F1-score of 0.89 and weighted F1-score of 0.95 across all continents (Supplementary Table 27).

City Classification City classification proved similarly successful, with both XGBoost and LightGBM exceeding 91% accuracy in cross-validation. The final meta-model achieved a test accuracy of 93%, macro F1-score of 0.80, and weighted F1-score of 0.92, representing a substantial improvement over all neural approaches (Supplementary Table 31). This high accuracy at both continent and city levels provides a strong foundation for accurate coordinate prediction.

Coordinate Regression and Geodesic Error For coordinate regression the R^2 was 0.896 and 0.8411 before and after coordinate transformation respectively. The ensemble leveraged TabPFN, which achieved exceptional geospatial precision. The test set median distance error was just 13.72 km, with a mean distance error of 589.02 km and a 95th percentile error of 3,577.48 km. Table 1 (Supplementary Table 35) provides a detailed analysis of error distribution across prediction groups.

When both continent and city predictions are correct (90.3% of cases), the median error drops dramatically to just 12.3 km.

Figure 7 visualizes the true and predicted coordinates for all test samples. The close alignment between blue (true) and red (predicted) points illustrates the high spatial accuracy achieved by the ensemble model across the globe.

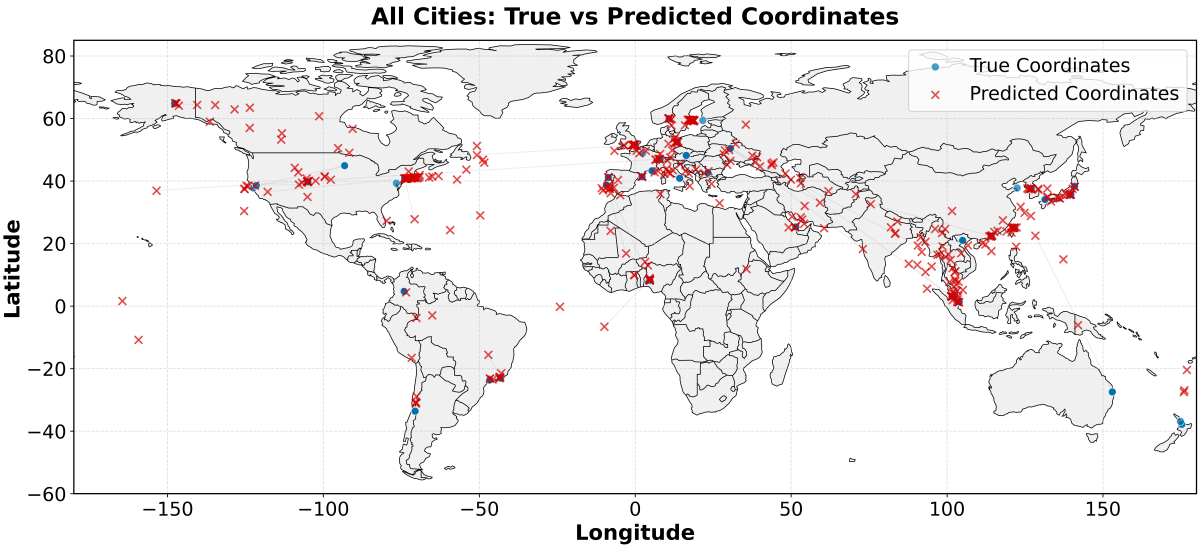

Figure 7. *World map showing the distribution of true coordinates (blue) and predicted coordinates (red) for test samples. The close alignment between true and predicted points illustrates the high spatial accuracy of the ensemble model.*

The distribution of geodesic errors by continent and city (Figure 8) shows that most predictions fall within small distance bins, especially for well-represented regions (Supplementary Table 27). This highlights the model's ability to achieve high spatial precision for the majority of test samples.

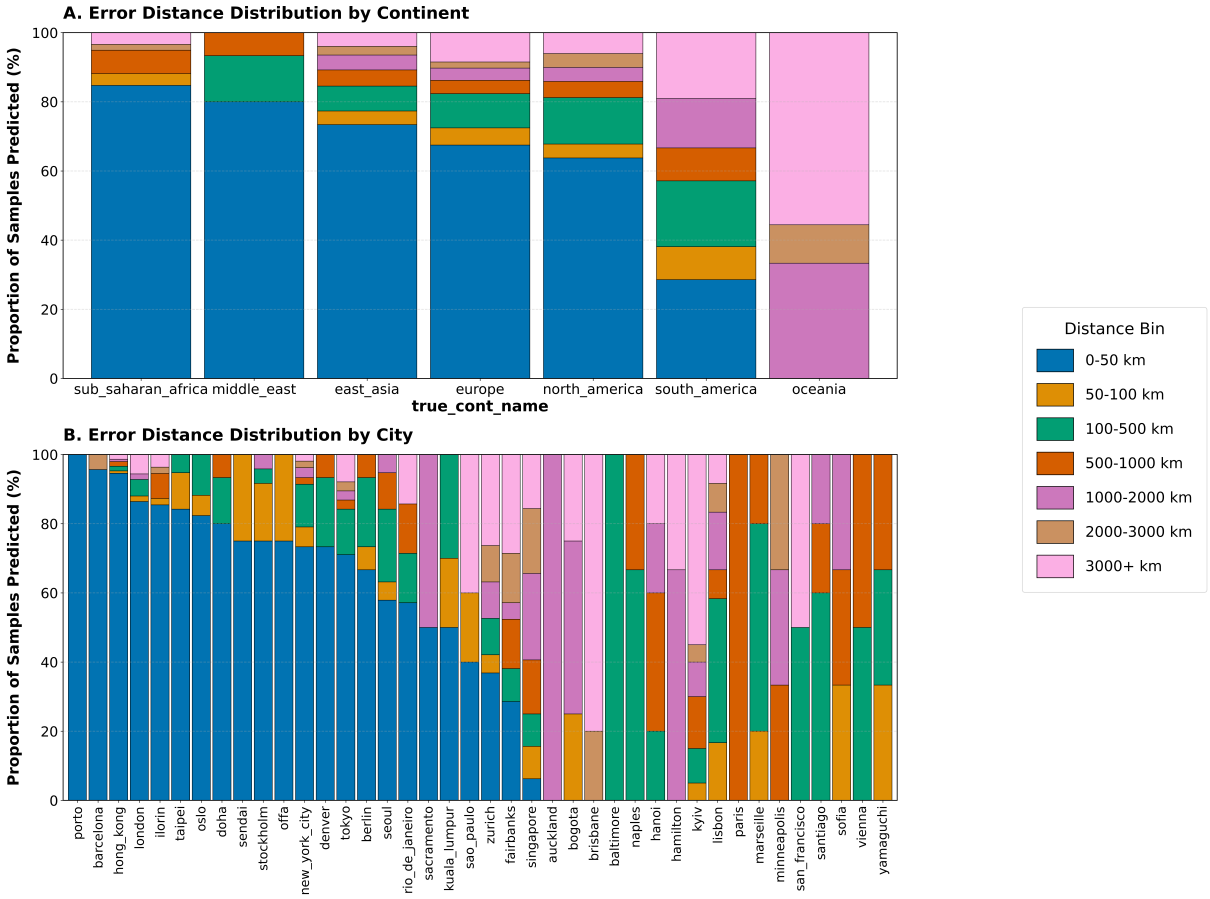

Figure 8. *Distribution of geodesic errors by continent and city for the ensemble model, showing the percentage of samples falling within various distance bins. Most predictions demonstrate high accuracy, especially for well-represented regions.*

In-Radius Accuracy The in-radius accuracy metrics in Supplementary Table 36 further demonstrate the ensemble model's precision. Around, 68.6% of predictions were within just 50 km of the true location, and 86.6% were within 1,000 km. These results outperform all neural network-based approaches and represent a significant increase in metagenomic geographic prediction.

3.5 Error Analysis and Hierarchical Propagation

Error group analysis for the ensemble learning model (Table 1) provides a clear understanding of how errors propagate through the prediction hierarchy (Liu et al., 2025). When both continent and city are correctly classified (Cc-Zc), the geodesic error is dramatically lower (e.g., median 12.3 km and mean 208.1 km for the ensemble model). However, errors at the continent or city level lead to a substantial increase in geodesic error (e.g., mean error 2148.1 km for Cc-Zi, 3902.2 km for Ci-Zc, and 7365.5 km for Ci-Zi), highlighting the importance of accurate hierarchical classification for precise coordinate prediction. This underscores the need for robust models at each level of the hierarchy to minimize overall geospatial error.

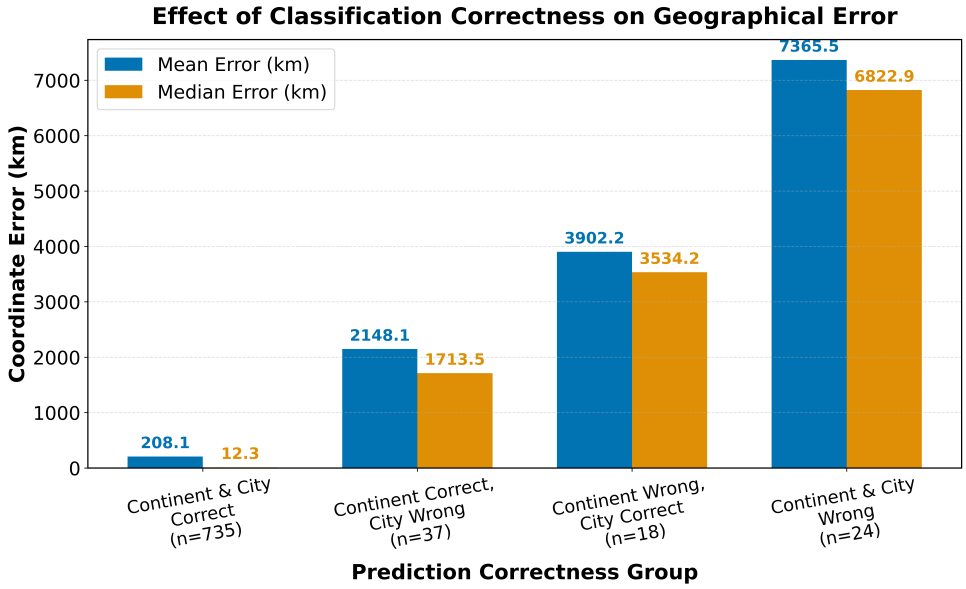

Figure 9. Classification correctness vs. geodesic error for ensemble model. The figure demonstrates the clear relationship between classification accuracy and coordinate prediction precision, with correctly classified samples showing dramatically lower geodesic errors.

3.6 Comparison with Previous State-of-the-Art (mGPS)

The mGPS (microbiome geographic population structure) tool (Zhang et al., 2024) represents the previous state-of-the-art for predicting the geographical origins of metagenomic samples from the MetaSUB dataset (Danko et al., 2021). Table 2 presents a comprehensive comparison between mGPS and our ensemble model across key performance metrics.

Table 2. Comparison of Ensemble Model and mGPS on MetaSUB Dataset

Metric	mGPS	Ensemble	Notes	Reference
Sample Size	4,070 (40 cities)	4,070 (40 cities)	After QC, matched setup	–
City Prediction Accuracy	92%	93%	Test set	Supplementary Table 31
Sensitivity	78%	86.6% (Continent), 81.1% (City)	Macro-average (see Supplementary)	Section 3.4.4
Specificity	99%	91.7% (Continent), 85.4% (City)	Macro-average (see Supplementary)	Section 3.4.4
In-Radius Accuracy				
<250 km	62%	77.27%	Proportion of predictions within 250 km	Table 1
<500 km	74%	81.94%	Proportion of predictions within 500 km	Table 1
<1,000 km	84%	86.61%	Proportion of predictions within 1,000 km	Table 1
Median Error (km)	137	13.72	Median geodesic error (km)	Table 1
AUC (Continent/City)	0.99–0.996	0.928 / 0.905	OVA/OVO macro-average ROC AUC	–
AUPR (Continent/City)	0.97 / 0.87	0.952 / 0.926	Macro-average precision-recall	–

Notes: mGPS and Ensemble models were evaluated on the same MetaSUB dataset after quality control. City prediction accuracy, sensitivity, and specificity are reported as macro-averages on the test set. In-radius accuracy indicates the proportion of predictions within the specified geodesic distance from the true location. Median error is the median geodesic distance between predicted and true coordinates. AUC and AUPR are reported as macro-averages for continent and city classification tasks. Bold values indicate superior performance.

The ensemble model achieved a city-level accuracy of 93%, slightly surpassing mGPS (92%). More

notably, it reduced the median coordinate error from 137 km (mGPS) to 13.72 km—a tenfold reduction—and increased the proportion of predictions within 250 km from 62% to 77.27%. The mean coordinate error was 589.02 km, and the 95th percentile error was 3577.48 km. While mGPS demonstrated slightly higher AUC values for classification tasks (0.99–0.996 vs. 0.928/0.905 for continent/city), our ensemble achieves comparable or superior AUPR scores (0.952/0.926 vs. 0.97/0.87 for continent/city), indicating strong performance even for imbalanced classes. Overall, our ensemble approach represents a significant advancement in the state of metagenomic geographic prediction, particularly in terms of coordinate precision and in-radius accuracy.

4. Discussion

Our hierarchical ensemble approach for geographic localization of metagenomic samples demonstrates significant advancements in prediction accuracy over existing methods. Below, we examine the implications of our findings, analyze model behaviors, and contextualize our work within the broader field.

4.1 Separate Neural Network Approach

While achieving reasonable continent classification performance (84.9% accuracy), the separate neural network approach revealed fundamental limitations in geographic localization tasks. The decrease in performance from continent to city level (from 84.9% to 70.1% accuracy) aligns with established machine learning principles that prediction difficulty increases with the number of target classes and granularity of distinctions (He and Garcia, 2009).

The most striking limitation was in coordinate prediction, where the median geodesic error of 4,237 km—nearly the width of the continental United States—indicates a fundamental inadequacy of independent networks for fine-grained spatial predictions. Regression tasks are generally more difficult than classification, especially in high-dimensional settings and with limited data (Caruana et al., 2008). This poor performance is due to the error propagation in hierarchical structures; early misclassifications cascade through the prediction pipeline with no mechanism for recovery or refinement (Liu et al., 2025). Such behavior demonstrates that metagenomic geographic signatures contain complex, interdependent spatial information that cannot be effectively captured by isolated models operating independently at different granularities. (Supplementary Table 24, 28, 36)

4.2 Combined Neural Network Approach

The dramatic improvement in coordinate regression accuracy achieved by our combined neural network approach (87.7% reduction in median error from 4,962 km to 1,631 km) highlights the critical importance of shared representations in hierarchical geographic tasks (Ruder, 2017). This finding has significant implications for metagenomic biogeography: it suggests that microbial communities contain spatial information at multiple scales that is best captured through multi-task learning frameworks that leverage shared patterns across different geographic resolutions.

The fact that regression accuracy improved much more than classification accuracy shows something important about metagenomic geographic signals. It suggests that microbial communities contain more information about continuous locations (like coordinates) than about broad categories (like continent or city). This fits with the idea that microbes spread gradually across regions, rather than following strict boundaries. Therefore, in future work, we should focus on breaking the continent into specific smaller regions that capture patterns based on factors such as average microbial signature changes from country to country. By generalizing to more classes at the continent level, we can potentially improve the granularity of predictions. However, precaution must be taken not to overdo this, as increasing the number of classes can introduce class imbalance and reduce overall model performance. (Supplementary Table 25, 29, 36)

4.3 GrowNet Model

The GrowNet results demonstrate the advantage of combining neural networks with gradient boosting principles for classification tasks. (Feng et al., 2021). The model's superior classification performance compared to traditional neural networks, yet inferior coordinate regression performance compared to the

combined neural network, reveals an important distinction in the types of geographic patterns present in microbiome data.

This performance pattern suggests that certain microbial features may serve as strong discriminative signals for categorical decisions (continent/city classification), while precise coordinate estimation requires modeling more subtle, continuous variations in community composition. This is due to the limited sample size, which can hinder the ability of boosting-based neural architectures to generalize in regression tasks (Zantvoort et al., 2024). (Supplementary Table 26, 30, 36)

4.4 Ensemble Learning

Our ensemble model's exceptional performance confirms findings from multiple domains that diverse models capturing different aspects of underlying patterns produce substantially more accurate predictions than any single approach (Opitz and Maclin, 1999). In the metagenomic context, this suggests that different algorithms are capturing complementary aspects of geographic signatures, potentially reflecting the complex, multi-faceted nature of microbial biogeography.

The superior performance of gradient boosting methods (XGBoost, LightGBM, CatBoost) for classification tasks aligns with recent research showing tree-based models often outperform deep learning on tabular data (Grinsztajn et al., 2022). This advantage likely stems from their ability to efficiently partition the feature space and model non-linear relationships without requiring extensive data or complex architectures—particularly valuable given the sparsity and high dimensionality characteristic of metagenomic data.

Interestingly, the transformer-based TabPFN model's superior performance for coordinate regression contradicts the general pattern favoring tree-based models for tabular data (Hütter et al., 2022). This unexpected finding suggests that coordinate prediction may benefit from attention mechanisms and global context modeling, which can better capture complex spatial relationships in metagenomic data. This result provides empirical evidence that different modeling approaches may be optimal for different aspects of the geographic prediction task, further justifying our ensemble approach.

Compared to the previous state-of-the-art mGPS tool (Zhang et al., 2024), which relied solely on XGBoost—a gradient boosted decision tree algorithm—as its primary machine learning model, our approach introduces a substantially more sophisticated ensemble framework. The mGPS tool was limited by the inductive biases and feature partitioning capabilities of a single model type, which, while effective for certain tasks, could not fully exploit the diverse patterns present in metagenomic data.

In contrast, our hierarchical ensemble leverages multiple model types, including neural networks, gradient boosting machines, and transformer architectures, and combines their strengths through meta-model learning. This approach allows each base model to specialize in particular aspects of the prediction task, such as continent, city, or precise coordinate localization. By integrating the outputs of these diverse models, the ensemble meta-model can more effectively capture both broad and subtle geographic signals, resulting in a significant boost to average F1 scores across all classes (Opitz and Maclin, 1999).

Most notably, our ensemble achieves a tenfold reduction in median coordinate error compared to mGPS, lowering the error from 137 km to 13.72 km. This leap in precision is largely attributable to the inclusion of advanced models such as transformers, which excel at capturing fine-grained variations in microbial signatures that are critical for pinpoint geographic localization. The transformer's attention mechanism enables the model to discern subtle shifts in microbial community composition, which traditional tree-based models may overlook.

This dramatic improvement in localization accuracy transforms the practical utility of metagenomic geographic prediction. Where previous methods could only assign samples to broad regions, our ensemble can now distinguish origins at the level of neighborhoods or districts within cities. Such high-resolution

attribution opens new possibilities for forensic microbiology, biosurveillance, and epidemiological investigations, where precise geographic information is essential for tracking sources and understanding microbial dispersal patterns (Robinson et al., 2021).

It is important to note that these results were obtained without any hyperparameter tuning; with further optimization, we expect performance to improve. (Supplementary Table 27, 31, 36)

4.5 Future Work and Limitations

Our results carry several important implications for predicting microbiome geography. Most notably, they reveal that microbiomes encode much finer geographic information than previously appreciated, challenging conventional views on microbial community assembly and biogeography. The high accuracy achieved by our models suggests that distinct geographic signatures exist even at small spatial scales, likely influenced by subtle environmental factors, human activity, or patterns of microbial dispersal, as seen in global urban microbiome surveys (Danko et al., 2021).

Additionally, the marked improvement of our ensemble approach over previous methods underscores the value of methodological innovation in maximizing the information extracted from metagenomic data. As sequencing technologies advance and datasets expand, ensemble strategies are likely to deliver even greater gains by making better use of larger sample sizes—a trend observed in other areas of machine learning (Caruana et al., 2008).

Looking ahead, incorporating temporal data into geographic prediction models represents a promising direction. Because microbiomes are dynamic and respond to seasonal and environmental changes, models that account for these temporal patterns could further enhance prediction accuracy and shed light on the stability of geographic signatures over time.

Despite these advances, our ensemble models do have limitations. Chief among them is the substantial computational demand, particularly in terms of GPU resources and runtime, which may restrict scalability and accessibility for researchers without high-performance computing infrastructure.

Future research should prioritize more robust and informative feature selection. Integrating additional biological knowledge—such as modeling interactions between microbial species—could provide deeper insights into the ecological processes underlying geographic signatures. Techniques like autoencoders may help extract more compact and meaningful representations from high-dimensional data. Further improvements could also be realized by diversifying the models within the ensemble, systematically optimizing hyperparameters, and leveraging domain expertise for feature engineering. Collectively, these strategies aim to improve both the interpretability and predictive accuracy of geographic models for metagenomic data.

4.6 Acknowledgements

I would like to thank my supervisor, Eran Elhaik, for his guidance and support throughout this project. I am also grateful to Bijan Mousavi and Sreejith for their valuable input and assistance during the course of this work.

4.7 Data and Code Availability

All data and source code supporting the findings of this study are publicly available at https://github.com/ChandrashekarCR/mgps_optimization. This repository includes the full implementation of our hierarchical ensemble models, preprocessing scripts, and instructions for reproducing the experiments. The metagenomic datasets used for training and evaluation are provided.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631.

Aydin, C. C., Demir, C., and Yilmaz, E. (2016). Capability of artificial neural network for forward conversion of geodetic coordinates (phi, lambda, h) to cartesian (x,y,z) coordinates. *Environmental Earth Sciences*, 75(7):1–10.

Bergman, A. (2025). Optimizing the microbial global population structure (mgps). Unpublished manuscript, cited with permission from the author.

Caruana, R., Karampatziakis, N., and Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 96–103, New York, NY, USA. Association for Computing Machinery.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Danko, D., Bezdan, D., Afshin, E. E., Ahsanuddin, S., Bhattacharya, C., Butler, D. J., Chng, K. R., Donnellan, D., Hecht, J., Jackson, K., Kuchin, K., Karasikov, M., Lyons, A., Mak, L., Meleshko, D., Mustafa, H., Mutai, B., Neches, R. Y., Ng, A., Nikolayeva, O., Nikolayeva, T., Png, E., Ryon, K. A., Sanchez, J. L., Shaaban, H., Sierra, M. A., Thomas, D., Young, B., Abudayyeh, O. O., Alicea, J., Bhattacharyya, M., Blekhman, R., Castro-Nallar, E., Cañas, A. M., Chatziefthimiou, A. D., Crawford, R. W., De Filippis, F., Deng, Y., Desnues, C., Dias-Neto, E., Dybwad, M., and Elhaik, E. (2021). A global metagenomic map of urban microbiomes and antimicrobial resistance. *Cell*, 184(13):3376–3393.e17.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple Classifier Systems*, 1857:1–15.

Feng, J., Wang, Y., Wang, Y., Wang, Y., and Liu, Y. (2021). Grownet: Refuel boosting with concatenation and forward propagation. In *Advances in Neural Information Processing Systems*, volume 34, pages 22237–22249.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data?

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Hütter, F., Zimmer, L., Probst, P., Hees, J., Krämer, N., and Hutter, F. (2022). Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pages 3146–3154.

Kosmopoulos, A., Partalas, I., Gaussier, E., Paliouras, G., and Androutsopoulos, I. (2014). Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Liu, H., Li, P., Hu, X., Bai, S., and Lin, Y. (2025). Multi-granularity decision information integration network for hierarchical classification via local and global constraints. *Applied Intelligence*, 55.

Mahdavi-Shahri, A., Houshmand, M., Yaghoobi, M., and Jalali, M. (2016). Applying an ensemble learning method for improving multi-label classification performance. In *2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)*, page 1–6. IEEE.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31:6638–6648.

Robinson, J. M., Pasternak, Z., Mason, C. E., and Elhaik, E. (2021). Forensic applications of microbiomics: A review. *Frontiers in Microbiology*, Volume 11 - 2020.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks.

Snyder, J. P. (1987). *Map Projections—A Working Manual*. U.S. Geological Survey Professional Paper 1395. U.S. Government Printing Office, Washington, DC.

Tang, L. (2024). Comparison the performances for distributed machine learning: Evidence from xgboost and dnn. *Applied and Computational Engineering*, 103:209–215.

Zantvoort, K., Nacke, B., Görlich, D., Hornstein, S., Jacobi, C., and Funk, B. (2024). Estimation of minimal data sets sizes for machine learning predictions in digital mental health interventions. *npj Digital Medicine*, 7(1):361.

Zhang, Y., McCarthy, L., Ruff, E., and Elhaik, E. (2024). Microbiome geographic population structure (mgps) detects fine-scale geography. *Genome Biology and Evolution*, 16(11):evae209.

5. Supplementary Materials

5.1 Supplementary Tables

5.1.1 Separate Neural Network Parameters

Table 3. 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

This table summarizes the architecture and training settings for each separate neural network used in the hierarchical pipeline. For each prediction stage (continent, city, coordinates), the hidden layers column specifies the number and size of fully connected layers, dropout indicates the range of dropout rates applied to reduce overfitting, batch normalization (Batch Norm) shows whether normalization was used, learning rate is the optimizer step size, batch size is the number of samples per training batch, and epochs is the total number of training iterations. These settings were selected to balance model complexity and generalization.

Table 4. 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

Each row lists a parameter and its default value for the continent, city, and coordinate neural network models.

Table 5. 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

Each row lists a hyperparameter and the range or set of values explored during tuning.

5.1.2 Combined Neural Network Parameters

Table 6. Architecture and training parameters for Combined Neural Networks.

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

This table details the architecture and training parameters for each branch of the Combined Neural Networks model. Each branch (continent, city, coordinates) uses a specific set of hidden layers, dropout rates, and batch normalization settings. The learning rate column indicates the optimizer step size for each branch. These parameters enable the model to jointly learn hierarchical tasks while sharing feature representations and minimizing error propagation.

Table 7. 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

Parameters are grouped by architecture and training settings, with their default values for the combined neural network.

Table 8. 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

Each row lists a hyperparameter and the range or set of values explored during tuning for the combined neural network.

5.1.3 GrowNet Parameters

Table 9. 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

Each row lists a parameter and its default value for the hierarchical GrowNet model, grouped by architecture, boosting, training, and loss weights.

Table 10. 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)

Each row lists a hyperparameter and the range or set of values explored during GrowNet tuning, including hierarchical loss weights.

5.1.4 Ensemble Learning Model Parameters

5.1.5 XGBoost Parameters

Table 11. 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

Each row lists a parameter and its default value for XGBoost classification and regression models.

Table 12. 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

Each row lists a hyperparameter and the range or set of values explored during XGBoost tuning.

5.1.6 LightGBM Parameters

Table 13. 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

Each row lists a parameter and its default value for LightGBM classification and regression models.

Table 14. 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

Each row lists a hyperparameter and the range or set of values explored during LightGBM tuning.

5.1.7 CatBoost Parameters

Table 15. 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

Each row lists a parameter and its default value for CatBoost classification and regression models.

Table 16. 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)

Each row lists a hyperparameter and the range or set of values explored during CatBoost tuning.

5.1.8 GrowNet Parameters

Table 17. 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

Each row lists a parameter and its default value for GrowNet classification and regression models in the ensemble context.

Table 18. 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

Each row lists a hyperparameter and the range or set of values explored during GrowNet tuning in the ensemble context.

5.1.9 Neural Network (MLP) Parameters

Table 19. 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

Each row lists a parameter and its default value for neural network (MLP) classification and regression models in the ensemble context.

Table 20. 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

Each row lists a hyperparameter and the range or set of values explored during neural network (MLP) tuning in the ensemble context.

5.1.10 TabPFN Parameters

Table 21. TabPFN model configuration

Parameter	Value
Model	Pre-trained TabPFN
Hyperparameter tuning	Max time

Each row lists a parameter or configuration for TabPFN models.

5.1.11 XGBoost Meta-Model Parameters

Table 22. 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

This table describes the configuration parameters for the meta-models used in the ensemble learning pipeline. The meta-models (XGBoost classifiers) are trained on out-of-fold predictions from base models to optimally combine their outputs. Key parameters include the algorithm type, objective function, tree depth, learning rate, number of estimators, and subsampling rates, all of which influence the meta-model's ability to generalize and aggregate predictions effectively.

Table 23. Ensemble layer specifications and selection criteria.

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

This table outlines the structure of each layer in the hierarchical ensemble model. For each layer (continent, city, coordinates), it specifies the input features used, the selection threshold for including base models in the ensemble, and the meta-model employed for combining predictions. The coordinate layer selects only the best-performing model based on median geodesic error, rather than using a meta-model, to preserve strong individual predictions.

5.1.12 Continent Classification: Separate Neural Network

Table 24. 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

Each row lists continent-level classification metrics (precision, recall, F1-score, support) for the separate neural network model.

5.1.13 Contienent Classification: Combined Neural Network

Table 25. 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

Each row lists continent-level classification metrics for the combined neural network model.

5.1.14 Continent Classification: Hierarchical GrowNet

Table 26. 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

Each row lists continent-level classification metrics for the hierarchical GrowNet model.

5.1.15 Continent Classification: Ensemble Learning

Each row lists continent-level classification metrics for the ensemble learning approach.

Table 27. Continent Classification Report (Ensemble Learning)

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

5.1.16 City Classification: Separate Neural Network

Table 28. City-level classification report for Separate Neural Network on the test set.

City	Prec.	Rec.	F1	Sup.
auckland	0.00	0.00	0.00	1
baltimore	0.33	1.00	0.50	1
barcelona	0.96	1.00	0.98	23
berlin	0.50	0.93	0.65	15
bogota	0.67	0.50	0.57	4
brisbane	0.40	0.80	0.53	5
denver	0.54	0.87	0.67	15
doha	0.93	0.93	0.93	15
europe	0.59	0.83	0.69	12
fairbanks	0.50	0.24	0.32	21
hamilton	0.25	0.33	0.29	3
hanoi	0.75	0.60	0.67	5
hong_kong	0.98	0.86	0.92	148
ilorin	0.87	0.62	0.72	55
kuala_lumpur	0.69	0.90	0.78	10
kyiv	0.42	0.50	0.45	20
lisbon	0.38	0.25	0.30	12
london	0.91	0.64	0.75	125
marseille	0.80	0.80	0.80	5
minneapolis	1.00	0.33	0.50	3
naples	0.67	0.67	0.67	3
new_york_city	0.72	0.83	0.77	105
offa	0.10	0.50	0.17	4
oslo	0.52	0.94	0.67	17
paris	0.00	0.00	0.00	1
rio_de_janeiro	0.83	0.71	0.77	7
sacramento	0.50	1.00	0.67	2
san_francisco	0.25	0.50	0.33	2
santiago	0.83	1.00	0.91	5
sao_paulo	0.40	0.40	0.40	5
sendai	0.33	1.00	0.50	4
seoul	0.77	0.89	0.83	19
singapore	0.45	0.31	0.37	32
sofia	0.50	0.67	0.57	3
stockholm	0.64	0.29	0.40	24
taipei	0.76	1.00	0.86	19
tokyo	0.67	0.53	0.59	38
vienna	0.00	0.00	0.00	4
yamaguchi	0.00	0.00	0.00	3
zurich	0.46	0.58	0.51	19
accuracy			0.70	814
macro avg	0.55	0.62	0.55	814
weighted avg	0.75	0.70	0.71	814

Each row lists city-level classification metrics for the separate neural network model.

5.1.17 City Classification: Combined Neural Network

Table 29. City-level classification report for Combined Neural Network on the test set.

City	Prec.	Rec.	F1	Sup.
auckland	0.00	0.00	0.00	1
baltimore	0.00	0.00	0.00	1
barcelona	0.96	1.00	0.98	23
berlin	1.00	0.13	0.24	15
bogota	0.00	0.00	0.00	4
brisbane	0.00	0.00	0.00	5
denver	0.76	0.87	0.81	15
doha	0.70	0.93	0.80	15
europe	0.39	1.00	0.56	12
fairbanks	0.75	0.43	0.55	21
hamilton	0.00	0.00	0.00	3
hanoi	0.00	0.00	0.00	5
hong_kong	0.94	0.99	0.96	148
ilorin	0.76	0.95	0.85	55
kuala_lumpur	0.78	0.70	0.74	10
kyiv	1.00	0.05	0.10	20
lisbon	0.33	0.17	0.22	12
london	0.94	0.74	0.83	125
marseille	0.00	0.00	0.00	5
minneapolis	0.00	0.00	0.00	3
naples	0.00	0.00	0.00	3
new_york_city	0.70	0.91	0.79	105
offa	0.00	0.00	0.00	4
oslo	0.58	0.82	0.68	17
paris	1.00	1.00	1.00	1
rio_de_janeiro	0.57	0.57	0.57	7
sacramento	0.50	0.50	0.50	2
san_francisco	0.50	1.00	0.67	2
santiago	0.83	1.00	0.91	5
sao_paulo	0.43	0.60	0.50	5
sendai	1.00	0.25	0.40	4
seoul	0.85	0.89	0.87	19
singapore	0.43	0.75	0.55	32
sofia	0.00	0.00	0.00	3
stockholm	0.87	0.54	0.67	24
taipei	0.90	1.00	0.95	19
tokyo	0.63	0.71	0.67	38
vienna	0.00	0.00	0.00	4
yamaguchi	0.00	0.00	0.00	3
zurich	0.53	0.53	0.53	19
accuracy			0.75	814
macro avg	0.49	0.48	0.45	814
weighted avg	0.75	0.75	0.72	814

Each row lists city-level classification metrics for the combined neural network model.

5.1.18 City Classification: Hierarchical GrowNet

Table 30. City-level classification report for Hierarchical GrowNet on the test set.

City	Prec.	Rec.	F1	Sup.
auckland	0.00	0.00	0.00	3
baltimore	0.00	0.00	0.00	0
barcelona	1.00	0.95	0.97	19
berlin	0.64	0.93	0.76	15
bogota	1.00	0.50	0.67	2
brisbane	0.33	0.50	0.40	4
denver	0.62	0.62	0.62	13
doha	0.74	0.93	0.82	15
europe	0.76	0.72	0.74	18
fairbanks	0.32	0.39	0.35	18
hamilton	0.00	0.00	0.00	2
hanoi	0.38	1.00	0.55	3
hong_kong	0.97	0.86	0.91	179
ilorin	0.91	0.74	0.81	53
kuala_lumpur	0.85	0.92	0.88	12
kyiv	0.19	0.46	0.27	13
lisbon	0.26	0.31	0.29	16
london	0.88	0.76	0.82	123
marseille	0.71	1.00	0.83	5
minneapolis	0.25	1.00	0.40	1
naples	1.00	0.20	0.33	5
new_york_city	0.75	0.74	0.75	105
offa	0.00	0.00	0.00	6
oslo	0.77	0.85	0.81	20
paris	0.33	0.50	0.40	2
rio_de_janeiro	1.00	0.67	0.80	6
sacramento	1.00	0.67	0.80	6
san_francisco	0.56	0.83	0.67	6
santiago	0.80	0.80	0.80	5
sao_paulo	1.00	0.75	0.86	8
sendai	0.67	1.00	0.80	6
seoul	0.71	1.00	0.83	15
singapore	0.59	0.42	0.49	24
sofia	0.33	0.50	0.40	2
stockholm	0.88	0.85	0.87	27
taipei	0.87	1.00	0.93	13
tokyo	0.73	0.70	0.71	23
vienna	0.50	1.00	0.67	1
yamaguchi	0.33	0.33	0.33	3
zurich	0.71	0.59	0.65	17
accuracy			0.75	814
macro avg	0.61	0.65	0.60	814
weighted avg	0.79	0.75	0.76	814

Each row lists city-level classification metrics for the hierarchical GrowNet model.

5.1.19 City Classification: Ensemble Learning

Table 31. City-level classification report for Ensemble Learning on the test set.

City	Prec.	Rec.	F1	Sup.
auckland	0.33	1.00	0.50	1
baltimore	0.00	0.00	0.00	1
barcelona	1.00	1.00	1.00	23
berlin	0.94	1.00	0.97	15
bogota	1.00	0.75	0.86	4
brisbane	1.00	0.60	0.75	5
denver	0.94	1.00	0.97	15
doha	1.00	0.93	0.97	15
fairbanks	0.83	0.95	0.89	21
hamilton	1.00	0.67	0.80	3
hanoi	1.00	0.80	0.89	5
hong_kong	0.99	0.99	0.99	148
ilorin	0.98	0.93	0.95	55
kuala_lumpur	0.91	1.00	0.95	10
kyiv	0.58	0.70	0.64	20
lisbon	0.92	0.92	0.92	12
london	1.00	0.97	0.98	125
marseille	0.75	0.60	0.67	5
minneapolis	0.60	1.00	0.75	3
naples	0.67	0.67	0.67	3
new_york_city	0.95	0.97	0.96	105
offa	0.67	1.00	0.80	4
oslo	1.00	0.94	0.97	17
paris	0.00	0.00	0.00	1
porto	0.92	1.00	0.96	12
rio_de_janeiro	1.00	0.86	0.92	7
sacramento	1.00	1.00	1.00	2
san_francisco	0.67	1.00	0.80	2
santiago	1.00	1.00	1.00	5
sao_paulo	1.00	0.60	0.75	5
sendai	1.00	1.00	1.00	4
seoul	0.86	0.95	0.90	19
singapore	0.73	0.84	0.78	32
sofia	1.00	0.67	0.80	3
stockholm	0.96	1.00	0.98	24
taipei	0.90	1.00	0.95	19
tokyo	0.85	0.87	0.86	38
vienna	0.60	0.75	0.67	4
yamaguchi	0.00	0.00	0.00	3
zurich	0.91	0.53	0.67	19
accuracy			0.93	814
macro avg	0.81	0.81	0.80	814
weighted avg	0.93	0.93	0.92	814

Each row lists city-level classification metrics for the ensemble learning approach.

Table 32. 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

Guide: C = Continent, Z = City. Groups indicate correctness of continent and city predictions.

Each row lists error group statistics for coordinate regression using the separate neural network model.

5.1.20 Coordinate Regression: Combined Neural Network

Table 33. 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

C = Continent, Z = City. Groups indicate correctness of continent and city predictions.

Each row lists error group statistics for coordinate regression using the combined neural network model.

5.1.21 Coordinate Regression Metrics: Hierarchical GrowNet

Table 34. 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

C = Continent, Z = City. Groups indicate correctness of continent and city predictions.

Each row lists error group statistics for coordinate regression using the hierarchical GrowNet model.

5.1.22 Coordinate Regression Metrics: Ensemble Learning Model

Table 35. Ensemble Learning Model: Error Group Analysis

Group	Count	Mean Error (km)	Median Error (km)	Proportion	Weighted Error
C_correct Z_correct	735	208.13	12.33	0.9029	187.93
C_correct Z_wrong	37	2148.09	1713.46	0.0455	97.64
C_wrong Z_correct	18	3902.22	3534.17	0.0221	86.29
C_wrong Z_wrong	24	7365.53	6822.91	0.0295	217.17

Note: C = Continent, Z = City. Groups indicate correctness of continent and city predictions.

5.1.23 In-Radius Accuracy Metrics

Table 36. In-Radius Accuracy Metrics for Separate Neural Network, Combined Neural Network, Hierarchical GrowNet, and Ensemble Learning on the test set.

Radius	Separate NN (%)	Combined NN (%)	GrowNet (%)	Ensemble (%)
<1 km	0.00	0.00	0.00	0.00
<5 km	0.00	0.00	0.00	4.18
<50 km	0.00	0.37	0.98	68.55
<100 km	0.00	9.46	2.70	72.85
<250 km	0.00	30.34	12.78	77.27
<500 km	0.86	49.75	30.96	81.94
<1000 km	1.84	66.34	57.37	86.61
<5000 km	55.65	89.31	89.07	96.44

Each row lists the percentage of predictions within specified radius thresholds for each model.

5.2 Figures

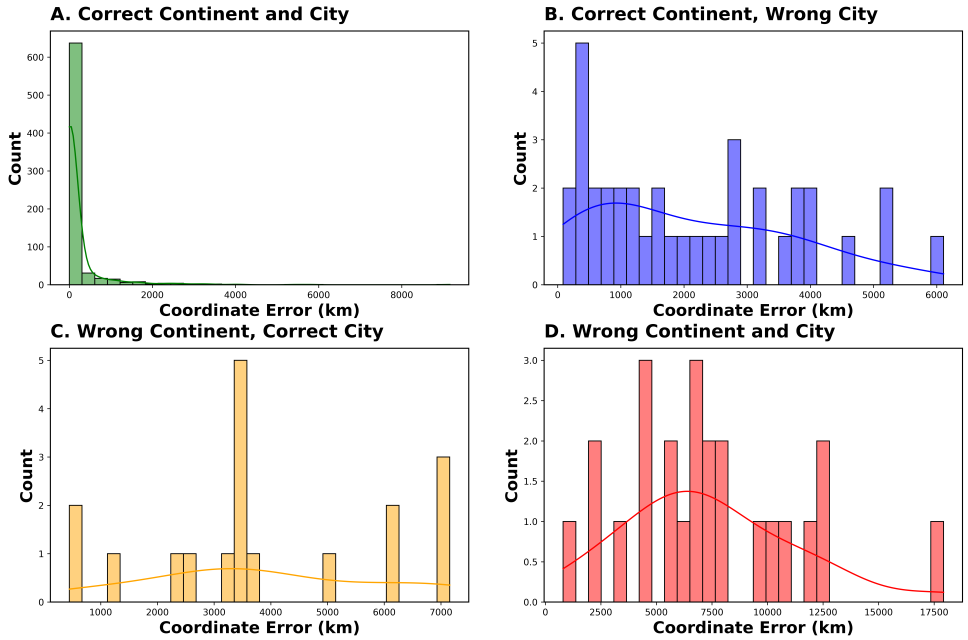

Figure 10. Error Distance Distribution by Continent and City for Ensemble Learning Model

This figure shows the distribution of error distances for the Ensemble Learning model, categorized by continent and city. The x-axis represents the error distance in kilometers, while the y-axis shows the frequency of occurrences. The figure highlights the performance of the model in predicting coordinates, with a focus on how errors propagate when the model predicts an incorrect continent or city.

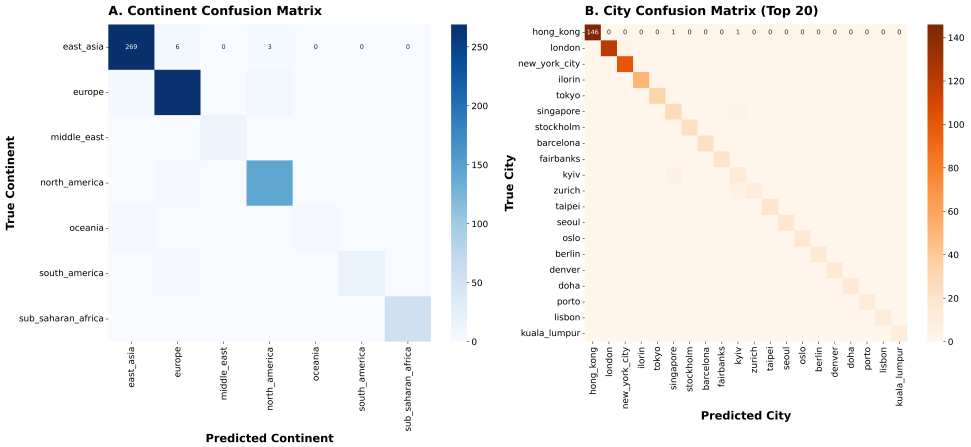

Figure 11. Confusion matrix for Continent and City Classification

This figure displays the confusion matrix for continent and city classification for the Ensemble Learning model. The x-axis represents the predicted classes, while the y-axis shows the true classes. Each cell indicates the number of instances classified into each category. The diagonal cells represent correct classifications, while off-diagonal cells indicate misclassifications. This matrix provides insights into the model's performance across different continents and cities, highlighting areas of strength and potential improvement. The confusion matrix is particularly useful for understanding how well the model distinguishes between different continents and cities