

mGPS Algorithm Optimization

**Course: Bioinformatics Research Project (BINP37),
15 credits**

Student: Chandrashekhar CR

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

Supervisor: Eran Elhaik

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

Lund University 2025

1 Abstract

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

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

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

26 **1. Introduction**

27 **1.1 Geographical Prediction Using Microbial Signatures**

28 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).

32 The microbial Global Population Structure (mGPS) algorithm 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).

38 **1.2 Previous Work and Methodology**

39 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 → city → coordinates, with prediction probabilities at each level used to inform subsequent predictions. (Zhang et al., 2024)

44 **1.3 Limitations in Existing Approaches**

45 Current approaches have several clear limitations. First, the hierarchical structure of prediction (continent → city → 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).

61 **1.4 Research Objectives and Contributions**

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

71 **1.5 Dataset and Proposed Improvements**

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

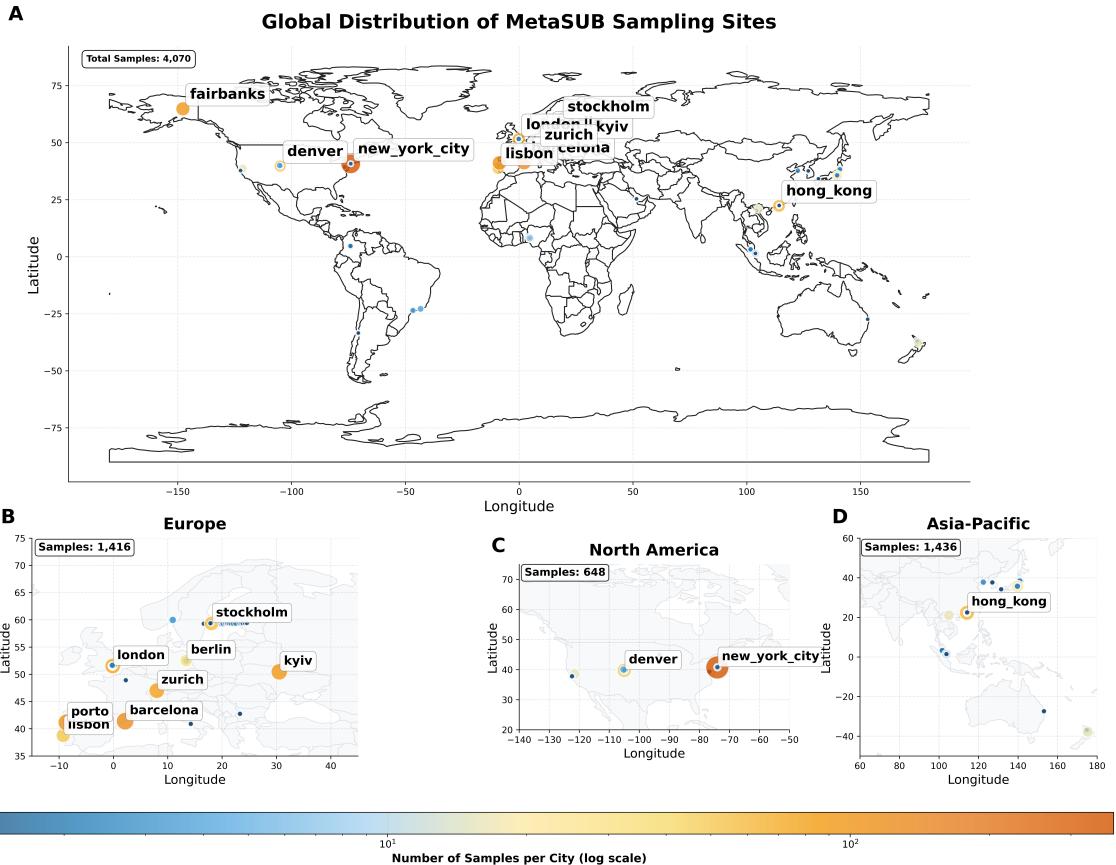


Figure 1. Global distribution of MetaSUB sampling sites. (A) World map showing sample locations and counts. (B-D) Regional breakdowns for Europe, North America, and Asia-Pacific. The color scale indicates the number of samples per city (log scale).

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

Taxonomic Diversity in MetaSUB Dataset
Analysis of 200 microbial species across 4070 metagenomic samples

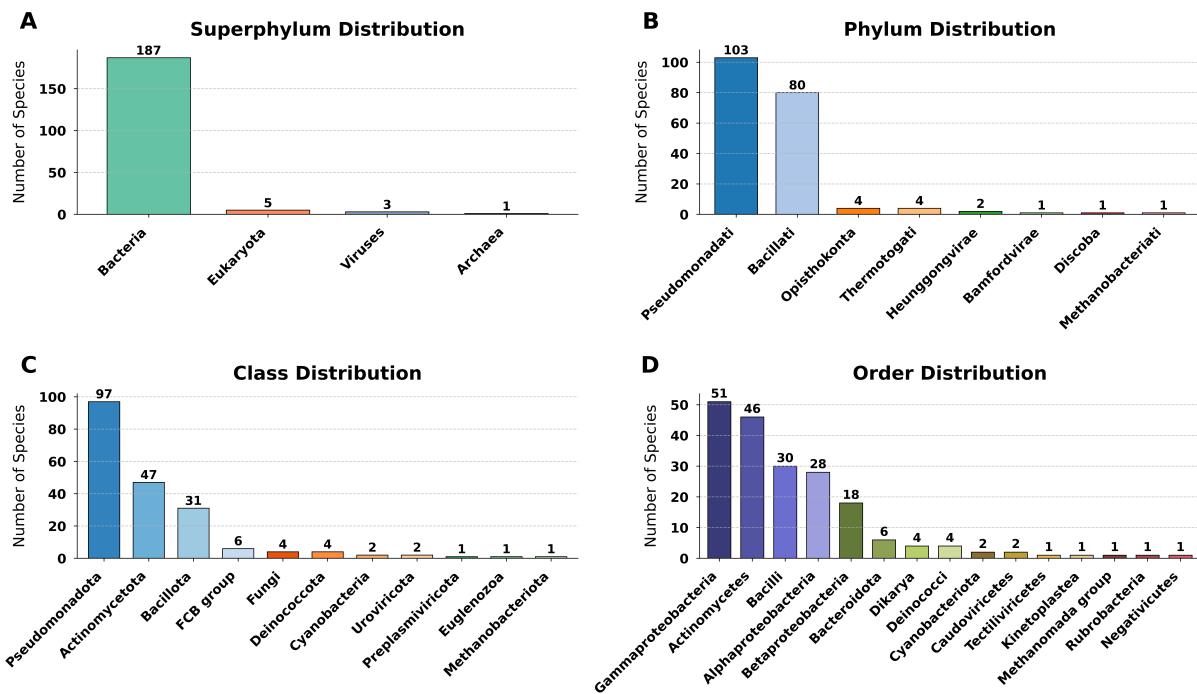


Figure 2. Taxonomic diversity in the MetaSUB dataset. (A) Superphylum, (B) Phylum, (C) Class, and (D) Order distributions for 200 microbial species. The dataset originally included 4,728 samples before quality control, and 4,070 post-QC samples from 40 cities were used for analysis. Bacteria dominate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

89 **2. Materials and Methods**

90 **2.1 Dataset and Preprocessing**

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

97 To maintain methodological consistency with previous mGPS work, we applied the
98 same quality control and feature selection procedures. Specifically, cities with fewer than
99 eight samples were excluded, and recursive feature elimination (RFE) with Random For-
100 est was used to reduce the initial set of approximately 3,000 microbial features to the
101 200–300 most informative, using 5-fold cross-validation (Guyon et al., 2002). Class imbal-
102 ance—particularly for underrepresented continents such as Oceania and Africa—was ad-
103 dressed using the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al.,
104 2002), achieving a 1:3 ratio between minority and majority classes. These steps ensured
105 that our dataset and preprocessing pipeline remained directly comparable to the original
106 mGPS study.

107 **2.2 Model Development**

108 We developed several modeling approaches to address the hierarchical geographic predic-
109 tion problem, each offering distinct advantages and characteristics.

110 **2.2.1 Neural Networks**

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

118 **Separate Neural Network Models** In accordance with the previous study, which
119 utilized a hierarchical approach with XGBoost (Zhang et al., 2024)(Chen and Guestrin,
120 2016), we constructed a set of independent neural networks to serve as baselines and
121 to analyze error propagation at each prediction level. Specifically, we developed three
122 specialized models: (1) a Continent Network that predicts continent labels from microbial

123 features; (2) a City Network that incorporates both microbial features and continent
124 probabilities to predict city labels; and (3) a Coordinate Network that leverages microbial
125 features, continent, and city probabilities to perform coordinate regression.

126 Default parameters and the hyperparameter search space for these models are provided
127 in Supplementary Tables 4 and 5.

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

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

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

138 For evaluation, we apply the inverse transformation to the predicted (x, y, z) values, con-
139 verting them back to latitude and longitude in radians, and then to degrees. This allows
140 for accurate geodesic error computation and ensures that the model predictions are inter-
141 pretable in the original coordinate system.

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

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

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

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

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

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

Separate Neural Networks Architecture

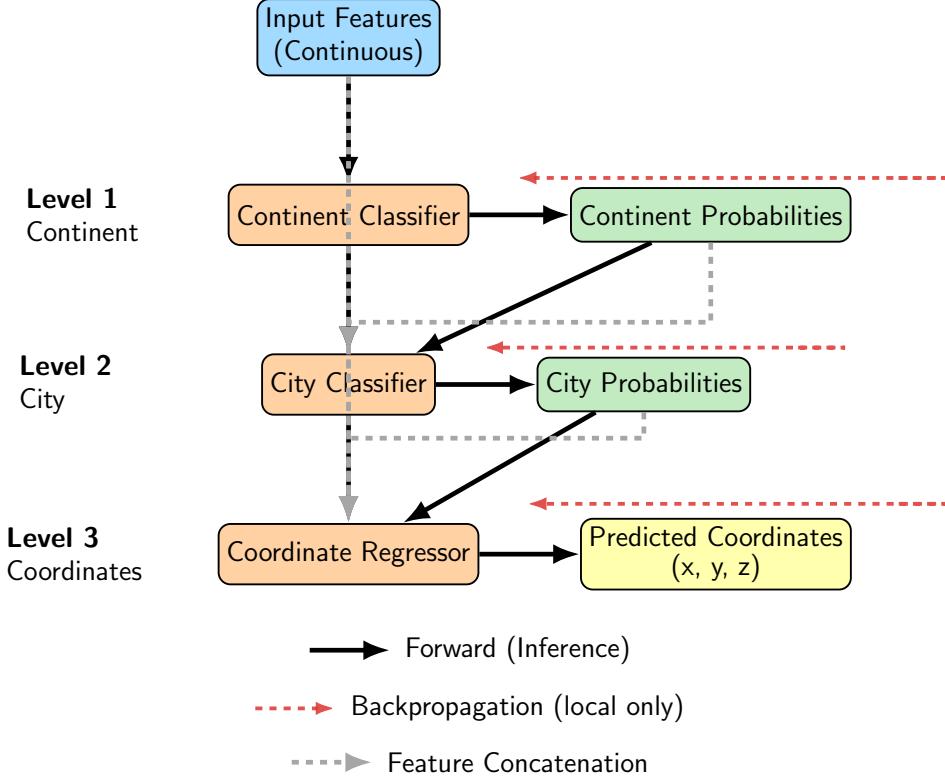


Figure 3. Schematic of the separate neural network approach for hierarchical geographic prediction. Each prediction level (continent, city, coordinates) is modeled by an independent neural network. Outputs from each level are used as inputs for the next, but training and backpropagation are performed independently for each network.

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

154 **Combined Neural Networks** To enable end-to-end hierarchical learning, we developed
 155 the Combined Neural Networks, a unified multi-task neural network architecture
 156 with three sequential branches. This model shares feature representations across tasks
 157 while maintaining task-specific output heads. Training is performed using a weighted
 158 multi-task loss, combining cross-entropy for classification tasks and mean squared error
 159 (MSE) for coordinate regression. As with the separate models, coordinate prediction in
 160 this architecture also employs the Cartesian transformation described in Equation 1 (Snyder,
 161 1987; Aydin et al., 2016).

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

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

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

165 where w_1, w_2, w_3 are the task-specific weights. This joint optimization strategy encourages
 166 the model to learn representations that are robust to error propagation by penalizing
 167 errors at higher levels more strongly, reflecting the hierarchical structure of the problem.
 168 During backpropagation, gradients flow through all branches, but their magnitudes are
 169 modulated by these weights, promoting robust feature learning across the hierarchy.

170 The architecture and training parameters for the Combined Neural Networks are sum-
 171 marized in Supplementary Table 6.

Combined Neural Networks Architecture

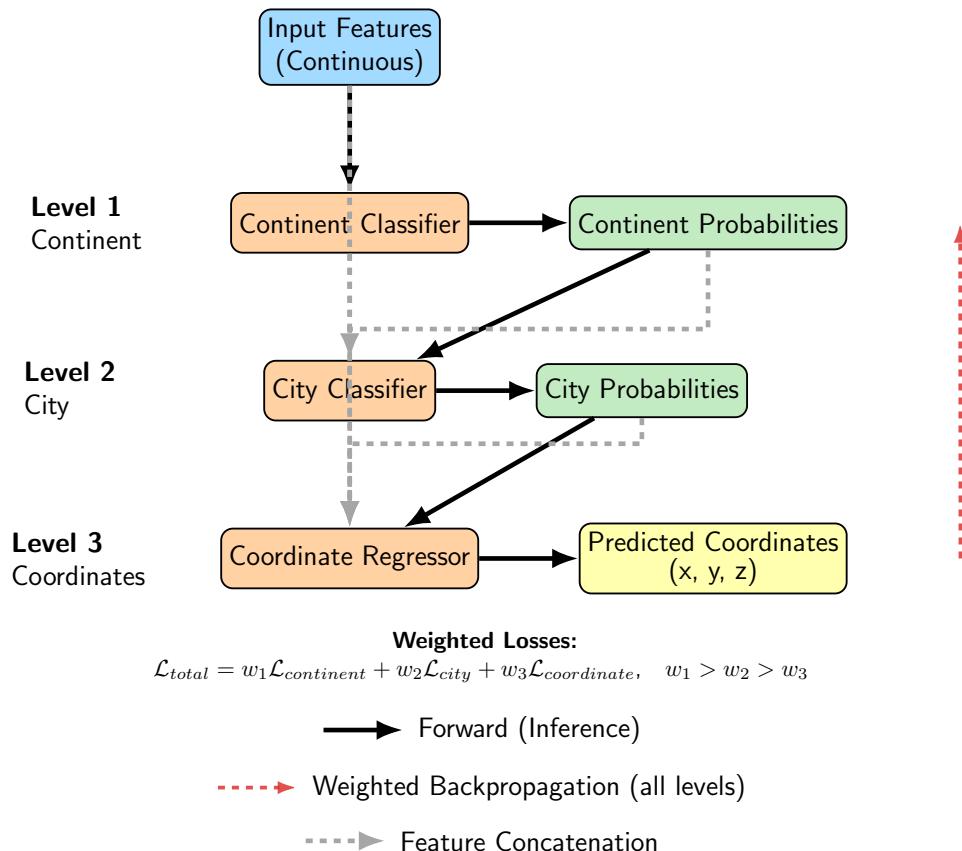


Figure 4. Diagram of the Combined Neural Networks architecture. This unified multi-task neural network consists of sequential branches for continent, city, and coordinate prediction. Feature representations are shared, and predictions from higher levels are concatenated with features for downstream tasks. 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.

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

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

182 2.2.2 GrowNet Architecture

183 We sought a model that could leverage the boosting principle—proven highly effective in
 184 tabular data settings by algorithms such as XGBoost (Chen and Guestrin, 2016)—while
 185 also benefiting from the flexibility and scalability of neural networks, which are known to
 186 perform better as dataset size increases (Tang, 2024). GrowNet (Feng et al., 2021) was
 187 chosen because it closely follows the boosting approach of XGBoost, but replaces decision
 188 trees with neural networks as weak learners. This design allows GrowNet to match or
 189 exceed the performance of leading tree-based models on tabular data, while providing
 190 improved adaptability for larger and more complex datasets. The selection of GrowNet
 191 was motivated by its algorithmic similarity to XGBoost and its demonstrated effectiveness
 192 in hierarchical, multi-task problems (Feng et al., 2021).

193 GrowNet is a gradient boosting framework that employs neural networks as weak
 194 learners for multi-task learning (Feng et al., 2021). The algorithm proceeds by sequentially
 195 adding shallow neural networks to the ensemble, each trained to correct the residuals
 196 (pseudo-residuals) of the previous learners, analogous to boosting in XGBoost (Chen
 197 and Guestrin, 2016). At each stage m , the pseudo-residuals $\mathbf{r}^{(m)}$ are computed as the
 198 negative gradient of the loss with respect to the current ensemble prediction, i.e., $\mathbf{r}^{(m)} =$
 199 $-\nabla_{F^{(m-1)}} \mathcal{L}(y, F^{(m-1)})$. Each weak learner h_m is then trained to fit these residuals.

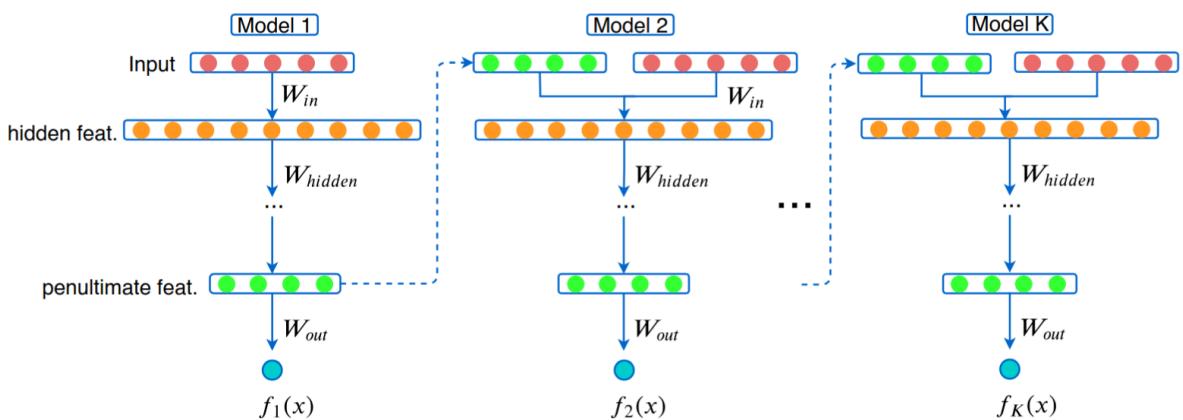


Figure 5. Diagram of the GrowNet architecture. This framework utilizes a multi-task learning approach with neural networks as weak learners, enabling effective handling of hierarchical tasks. Figure obtained from (Feng et al., 2021).

200 The hierarchical GrowNet training algorithm proceeds as follows:

201 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
202 stages), ρ (learning rate), λ (optimizer step size), and epochs_per_stage.

203 2. Initialize baseline predictions $F^{(0)}$.

204 3. For $m = 1$ to M :

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

206 (b) Initialize a new weak learner h_m .

207 (c) For each epoch in epochs_per_stage:

208 i. Sample a mini-batch B .

209 ii. Compute gradients and update h_m parameters using $\nabla_{\theta} \mathcal{L}_{residual}(B; h_m)$.

210 (d) Update ensemble: $F^{(m)} = F^{(m-1)} + \rho \cdot h_m$.

211 (e) Periodically, jointly fine-tune all weak learners via corrective optimization:

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

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

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

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

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

227 2.2.3 Ensemble Learning

228 **Model Selection and Integration Strategy:** Our ensemble strategically combines
229 different models to minimize hierarchical error across varying data regimes: gradient
230 boosting models (XGBoost, LightGBM, CatBoost), TabPFN, neural networks (MLPs),
231 and GrowNet. This selection balances proven effectiveness on tabular data with scalability

232 for larger datasets, ensuring robust performance across different data scenarios (Chen and
233 Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018; Grinsztajn et al., 2022; Hütter
234 et al., 2022; Feng et al., 2021; Caruana et al., 2008; Tang, 2024; Dietterich, 2000; Opitz and
235 Maclin, 1999). The ensemble employs task-specific integration mechanisms: classification
236 tasks use threshold filtering with XGBoost meta-models (Supplementary Table 22) to
237 leverage diverse model strengths, while regression tasks select only the best-performing
238 single model to preserve granular predictions, as illustrated in Figure 6.

239 **Model Architecture and Hyperparameters:** The ensemble incorporates the fol-
240 lowing models. Gradient boosting models—including XGBoost (Chen and Guestrin,
241 2016) (see Supplementary Tables 11, 12), LightGBM (Ke et al., 2017) (Supplemen-
242 tary Tables 13, 14), and CatBoost (Prokhorenkova et al., 2018) (Supplementary Ta-
243 bles 15, 16)—are optimized for capturing non-linear relationships in tabular data. TabPFN (Hüt-
244 ter et al., 2022) is a prior-data fitted neural network leveraging meta-learning for rapid
245 adaptation to new tabular tasks (see Supplementary Table 21). Standard multilayer
246 perceptrons provide capacity for complex feature interactions at scale (Supplementary
247 Tables 19, 20). GrowNet (Feng et al., 2021) is a gradient boosting neural network archi-
248 tecture offering robust performance for larger datasets with intricate relationships (Sup-
249 plementary Tables 17, 18).

Hierarchical Ensemble Architecture

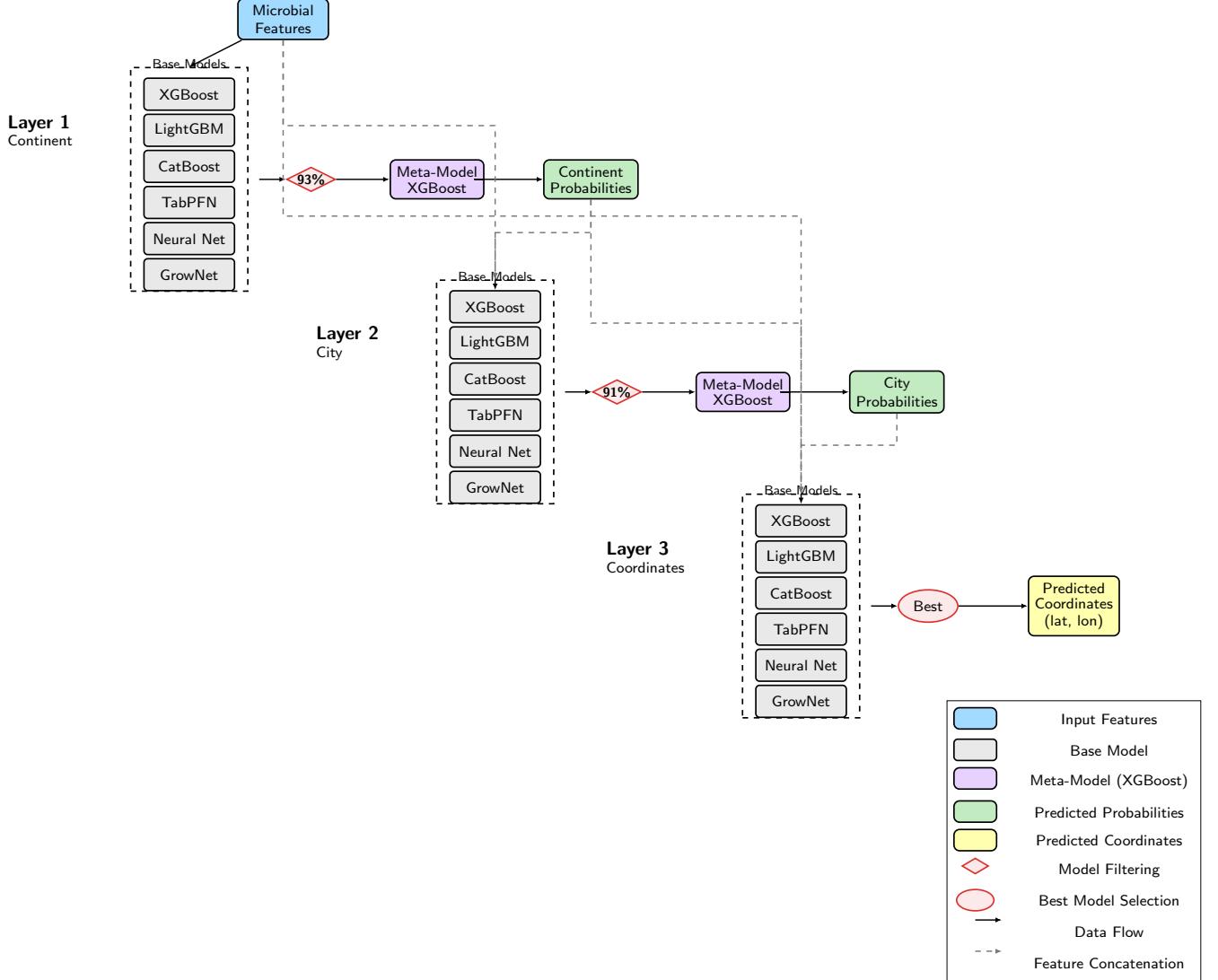


Figure 6. Overview of the hierarchical ensemble learning workflow. The ensemble is organized in three layers: continent classification, city classification, and coordinate regression. At each stage, predictions from multiple base models are combined using meta-models, and probability outputs are used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

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

259 configurations to find the optimal fit for the data (maximum of 300 seconds). For each
260 selected and tuned model, out-of-fold (OOF) predictions are generated using 5-fold cross-
261 validation: in each fold, the model is trained on $k - 1$ folds (with tuned hyperparameters)
262 and predicts on the held-out fold. This ensures that every OOF prediction is made by
263 a model that has not seen the corresponding sample during either training or hyperpa-
264 rameter selection, thus preventing information leakage. The concatenated OOF predic-
265 tions from all selected models are used as meta-features to train the meta-model (e.g.,
266 XGBoost), which learns to optimally combine the base models' outputs. For TabPFN
267 models that pass the threshold, we always use the maximum training time setting for
268 both OOF prediction generation and final model training to ensure optimal adaptation to
269 the dataset. Layer 2 (City Classification) builds on this by using both the original micro-
270 bial features and continent probabilities from Layer 1; models surpassing a 91% accuracy
271 threshold are included, and the same meta-learning protocol is followed to leverage the di-
272 verse inductive biases of different models, as some excel at predicting specific geographic
273 regions. Layer 3 (Coordinate Prediction) utilizes the complete feature set—microbial
274 abundances, continent probabilities, and city probabilities—but, unlike the classification
275 layers, selects only the single best-performing model for final predictions. This is because
276 averaging continuous regression outputs can degrade performance by smoothing strong
277 individual predictions (Dietterich, 2000; Opitz and Maclin, 1999). For coordinate regres-
278 sion, two approaches are evaluated: tree-based models predict latitude first, followed by
279 longitude conditioned on the predicted latitude, while neural networks directly predict
280 3D Cartesian coordinates (see Equation 1 (Snyder, 1987; Aydin et al., 2016)), which are
281 subsequently converted to latitude and longitude. The model achieving the lowest median
282 Haversine distance error is selected for final predictions. This dynamic selection mecha-
283 nism, depicted in Figure 6, allows the ensemble to adapt as datasets grow, transitioning
284 from tree-based models to neural networks when data volume increases (Tang, 2024).

285 **Feature Augmentation and Data Flow** At each stage of the hierarchical ensemble,
286 the input features are systematically augmented to improve prediction accuracy. The
287 initial features are the relative sequence abundances, representing the proportion of each
288 microbial species present in a sample. These undergo recursive feature elimination (RFE),
289 a process that selects the most informative features for the task, resulting in the matrix
290 X_{cont} used for continent prediction. Meta-models (such as XGBoost) are then trained
291 on the outputs of multiple base models to produce refined probability estimates for each
292 class. These probabilities are concatenated with the original features and used as inputs
293 for subsequent prediction stages (city and coordinates) (Supplementary Table 23). The
294 equations below summarize this process:

$$X_{cont} = \text{RFE}(X_{microbial}) \quad (6)$$

$$\hat{P}_{cont} = \text{MetaModel}_{cont}(\{f_i(X_{cont})\}_{i=1}^N) \quad (7)$$

$$X_{city} = [X_{cont}; \hat{P}_{cont}] \quad (8)$$

$$\hat{P}_{city} = \text{MetaModel}_{city}(\{f_j(X_{city})\}_{j=1}^M) \quad (9)$$

$$X_{coord} = [X_{cont}; \hat{P}_{cont}; \hat{P}_{city}] \quad (10)$$

$$\hat{Y}_{coord} = f_{best}(X_{coord}) \quad (11)$$

295 Here, X_{cont} is the reduced feature set after RFE, \hat{P}_{cont} and \hat{P}_{city} are the predicted
 296 probabilities from the meta-models for continent and city, and f_{best} denotes the best-
 297 performing regression model for coordinate prediction. This augmentation ensures that
 298 each prediction layer benefits from both the original microbial features and the most
 299 confident predictions from previous layers.

300 2.3 Error Propagation and Geodesic Error Calculation

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

$$E(D) = P_{cc,zc} E_{cc,zc} + P_{cc,zi} E_{cc,zi} + P_{ci,zc} E_{ci,zc} + P_{ci,zi} E_{ci,zi} \quad (12)$$

304 where:

- 305 • $P_{cc,zc} = P(C = C^*, Z = Z^*)$ is the probability of predicting both the correct
 306 continent and correct city,
- 307 • $P_{cc,zi} = P(C = C^*, Z \neq Z^*)$ is the probability of predicting the correct continent
 308 but incorrect city,
- 309 • $P_{ci,zc} = P(C \neq C^*, Z = Z^*)$ is the probability of predicting the incorrect continent
 310 but correct city,
- 311 • $P_{ci,zi} = P(C \neq C^*, Z \neq Z^*)$ is the probability of predicting both the incorrect continent
 312 and incorrect city,
- 313 • $E_{cc,zc} = E(D|C = C^*, Z = Z^*)$ is the expected geodesic error when both continent
 314 and city are correct,
- 315 • $E_{cc,zi} = E(D|C = C^*, Z \neq Z^*)$ is the expected error when continent is correct but
 316 city is incorrect,

317 • $E_{ci,zc} = E(D|C \neq C^*, Z = Z^*)$ is the expected error when continent is incorrect but
318 city is correct,

319 • $E_{ci,zi} = E(D|C \neq C^*, Z \neq Z^*)$ is the expected error when both continent and city
320 are incorrect.

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

323 **Geodesic Error Calculation (Haversine Formula)** Geodesic error is computed as
324 the great-circle distance between predicted and true coordinates using the Haversine for-
325 mula:

$$d = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right) \quad (13)$$

326 where:

- 327 • d is the geodesic distance (in kilometers),
328 • R is the Earth's radius (mean value $R = 6371$ km),
329 • ϕ_1, ϕ_2 are the latitudes (in radians) of the true and predicted points,
330 • λ_1, λ_2 are the longitudes (in radians) of the true and predicted points,
331 • $\Delta\phi = \phi_2 - \phi_1$,
332 • $\Delta\lambda = \lambda_2 - \lambda_1$.

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

335 **3. Results**

336 **3.1 Overview**

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

341 **3.2 Dataset and Evaluation Metrics**

342 We evaluated all models on the filtered MetaSUB dataset, containing 4,070 samples from
343 40 cities on 7 continents. Data were partitioned into training, validation, and test sets
344 (2,604/652/814 samples, respectively) after quality control. The dataset exhibits class
345 imbalance, particularly at the continent and city levels.

346 Principal metrics of evaluation are **classification accuracy**, **macro-averaged F1-**
347 **score**, and **weighted F1-score** for categorical predictions at both continent and city
348 scales. For geospatial accuracy estimation, we measured **geodesic error**, the great-circle
349 distance between predicted and actual coordinates on Earth's surface.¹³ We also provide
350 **in-radius accuracy** (the proportion of predictions within specified geodesic distances
351 of the true location). On classification tasks, **AUPR** (area under the precision-recall
352 curve) and **AUC** (area under the ROC curve) are only reported for the ensemble model
353 to facilitate a balanced comparison with the mGPS state-of-the-art model. (Zhang et al.,
354 2024)

355 **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.

R² (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.

356

357 **3.4 Model Performance**

358 This section presents the performance of the various models evaluated on the MetaSUB
 359 dataset, focusing on continent and city classification accuracy, geodesic error, and in-
 360 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

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

365 **3.4.1 Separate Neural Networks**

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

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

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

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

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

384 **3.4.2 Combined Neural Networks**

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

396 **3.4.3 Hierarchical GrowNet**

397 GrowNet, which combines neural networks with gradient boosting principles (Feng et al., 2021), achieved
398 the highest classification accuracy among neural models. It reached 86.4% continent accuracy (macro F1-
399 score: 0.77, weighted F1-score: 0.86; Supplementary Table 26) and 75.1% city accuracy (macro F1-score:
400 0.60, weighted F1-score: 0.76; Supplementary Table 30).

401 For coordinate regression, GrowNet achieved a median geodesic error of 823 km and mean error of
402 1,885 km, with a maximum error of 18,964 km. The coordinate regression R^2 was 0.685 and 0.627 before
403 and after inverse transformation of predicted coordinates to latitude and longitude respectively. The
404 in-radius accuracy was 57.4% within 1,000 km and 89.1% within 5,000 km (Supplementary Table 36).
405 Supplementary Table 34 provides a detailed error analysis by prediction group. Compared to both
406 separate and combined neural networks, GrowNet showed lesser performance in city prediction accuracy
407 to the combined neural network approach.

408 **3.4.4 Ensemble Learning Model**

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

415 **Continent Classification** The ensemble model achieved the highest continent classification ac-
416 curacy (95.0%) among all approaches. Even for underrepresented continents like Oceania, the model
417 maintained reasonable performance, with a macro-averaged F1-score of 0.89 and weighted F1-score of
418 0.95 across all continents (Supplementary Table 27).

419 **City Classification** City classification proved similarly successful, with both XGBoost and Light-
420 GBM exceeding 91% accuracy in cross-validation. The final meta-model achieved a test accuracy of
421 93%, macro F1-score of 0.80, and weighted F1-score of 0.92, representing a substantial improvement over
422 all neural approaches (Supplementary Table 31). This high accuracy at both continent and city levels
423 provides a strong foundation for accurate coordinate prediction.

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

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

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

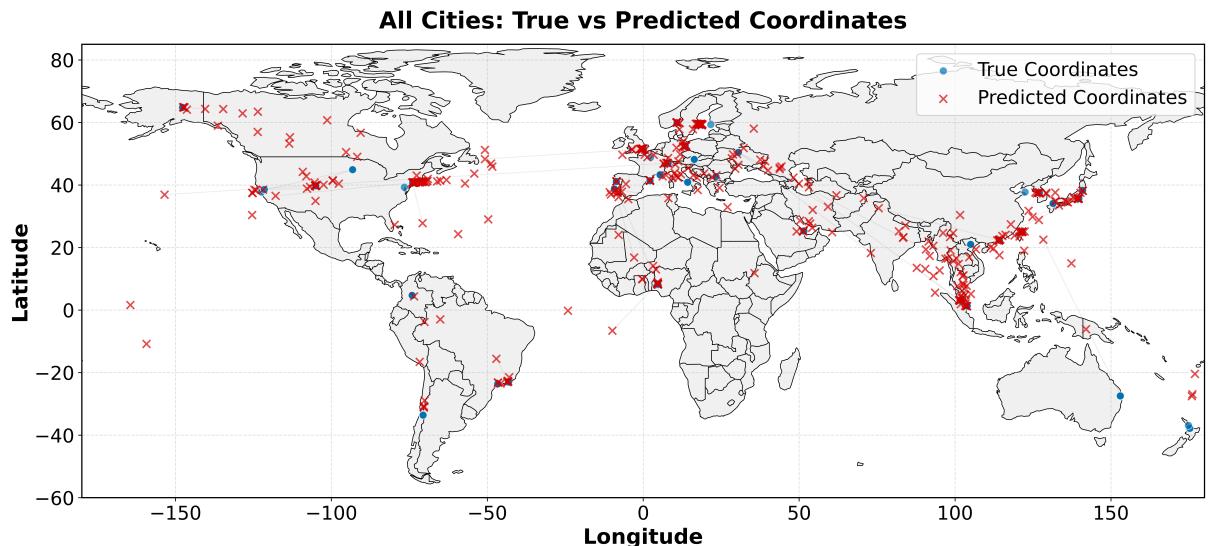


Figure 7. World map showing the distribution of true coordinates (blue) and predicted coordinates (red) for test samples. The close alignment between true and predicted points illustrates the high spatial accuracy of the ensemble model.

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

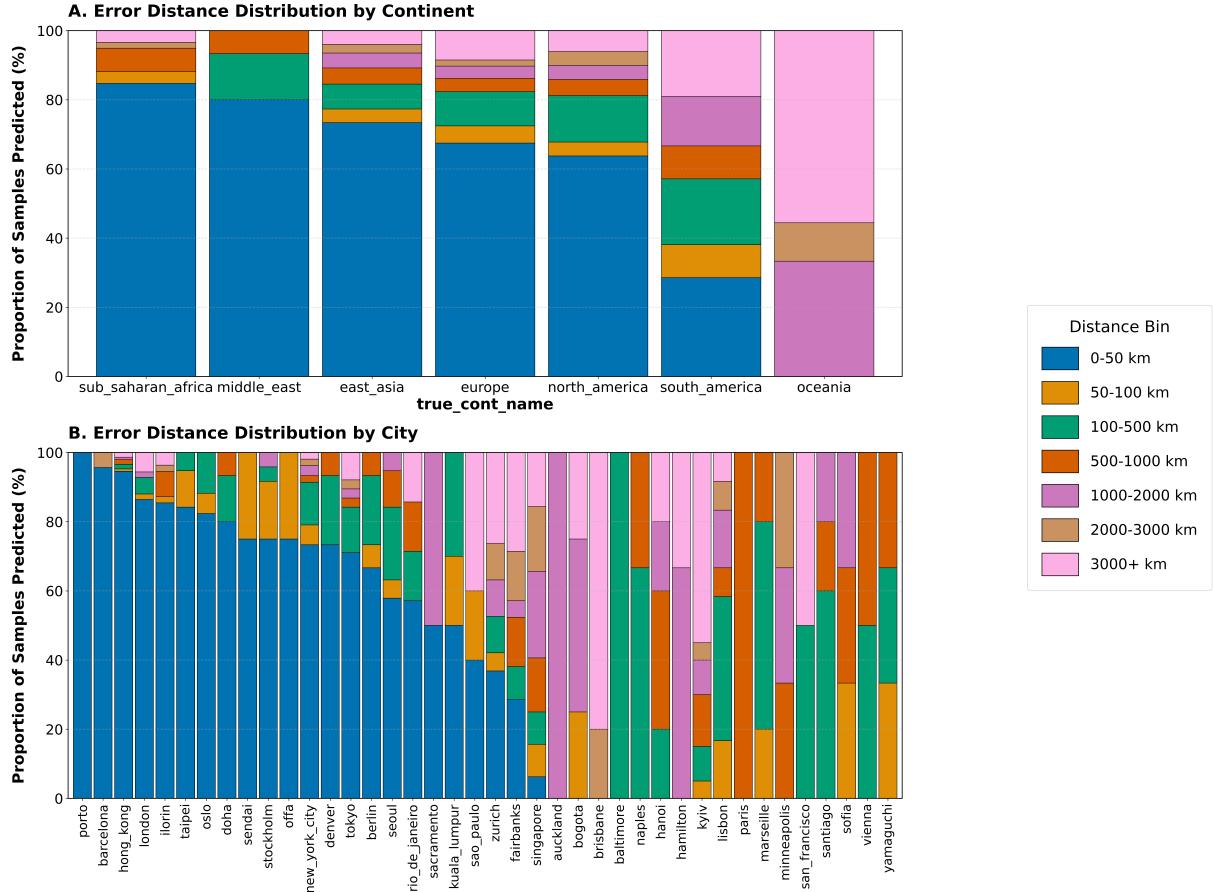


Figure 8. Distribution of geodesic errors by continent and city for the ensemble model, showing the percentage of samples falling within various distance bins. Most predictions demonstrate high accuracy, especially for well-represented regions.

437 **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 438 location, and 86.6% were within 1,000 km. These results outperform all neural network-based approaches 439 and represent a significant increase in metagenomic geographic prediction.

441 3.5 Error Analysis and Hierarchical Propagation

442 Error group analysis for the ensemble learning model (Table 1) provides a clear understanding of how 443 errors propagate through the prediction hierarchy (Liu et al., 2025). When both continent and city are 444 correctly classified (Cc-Zc), the geodesic error is dramatically lower (e.g., median 12.3 km and mean 208.1 445 km for the ensemble model). However, errors at the continent or city level lead to a substantial increase 446 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), 447 highlighting the importance of accurate hierarchical classification for precise coordinate prediction. This 448 underscores the need for robust models at each level of the hierarchy to minimize overall geospatial error.

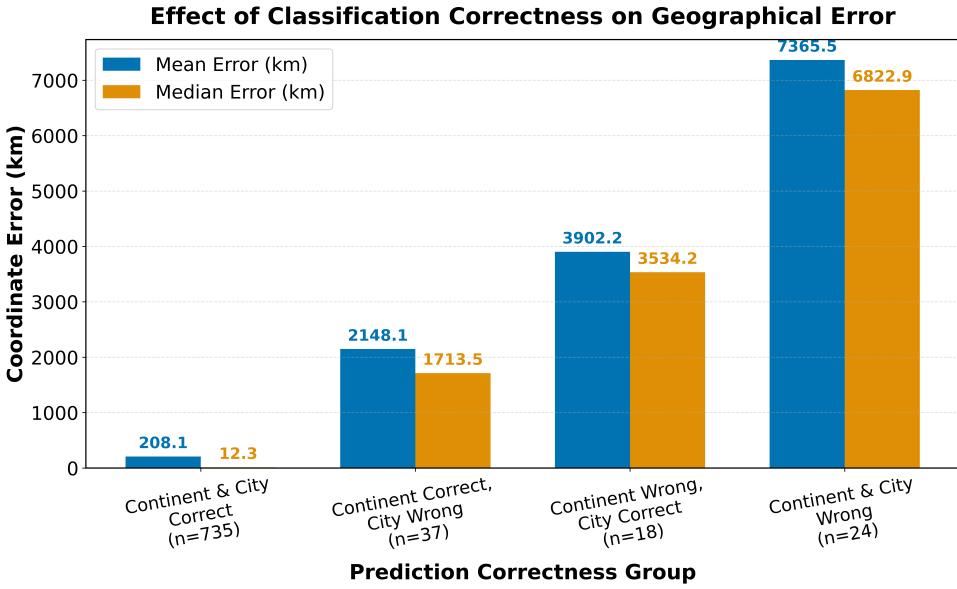


Figure 9. Classification correctness vs. geodesic error for ensemble model. The figure demonstrates the clear relationship between classification accuracy and coordinate prediction precision, with correctly classified samples showing dramatically lower geodesic errors.

449 3.6 Comparison with Previous State-of-the-Art (mGPS)

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

Table 2. 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%	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.

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

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

463 **4. Discussion**

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

467 **4.1 Separate Neural Network Approach**

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

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

482 **4.2 Combined Neural Network Approach**

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

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

499 **4.3 GrowNet Model**

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

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

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

510 4.4 Ensemble Learning

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

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

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

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

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

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

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

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

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

553 4.5 Future Work and Limitations

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

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

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

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

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

580 4.6 Acknowledgements

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

584 **References**

- 585 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.
- 588 Aydin, C. C., Demir, C., and Yilmaz, E. (2016). Capability of artificial neural network for forward conversion of geodetic coordinates (ϕ , λ , h) to cartesian (x, y, z) coordinates. *Environmental Earth Sciences*, 75(7):1–10.
- 591 Bergman, A. (2025). Optimizing the microbial global population structure (mgps). Unpublished manuscript, cited with permission from the author.
- 593 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.
- 596 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.
- 598 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.
- 600 Danko, D., Bezdan, D., Afshin, E. E., Ahsanuddin, S., Bhattacharya, C., Butler, D. J., Chng, K. R.,
601 Donnellan, D., Hecht, J., Jackson, K., Kuchin, K., Karasikov, M., Lyons, A., Mak, L., Meleshko, D.,
602 Mustafa, H., Mutai, B., Neches, R. Y., Ng, A., Nikolayeva, O., Nikolayeva, T., Png, E., Ryon, K. A.,
603 Sanchez, J. L., Shaaban, H., Sierra, M. A., Thomas, D., Young, B., Abudayyeh, O. O., Alicea, J.,
604 Bhattacharyya, M., Blekhman, R., Castro-Nallar, E., Cañas, A. M., Chatzifethimiou, A. D., Crawford,
605 R. W., De Filippis, F., Deng, Y., Desnues, C., Dias-Neto, E., Dybwad, M., and Elhaik, E. (2021). A
606 global metagenomic map of urban microbiomes and antimicrobial resistance. *Cell*, 184(13):3376–
607 3393.e17.
- 608 Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple Classifier Systems*, 1857:1–15.
- 609 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.
- 612 Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data?
- 614 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.
- 616 He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.
- 618 Hüttner, 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*.

- 620 Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A
621 highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*,
622 volume 30, pages 3146–3154.
- 623 Kosmopoulos, A., Partalas, I., Gaussier, E., Paliouras, G., and Androutsopoulos, I. (2014). Evaluation
624 measures for hierarchical classification: a unified view and novel approaches. *Data Mining and*
625 *Knowledge Discovery*, 29(3):820–865.
- 626 LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- 627 Liu, H., Li, P., Hu, X., Bai, S., and Lin, Y. (2025). Multi-granularity decision information integration
628 network for hierarchical classification via local and global constraints. *Applied Intelligence*, 55.
- 629 Mahdavi-Shahri, A., Houshmand, M., Yaghoobi, M., and Jalali, M. (2016). Applying an ensemble learning
630 method for improving multi-label classification performance. In *2016 2nd International Conference of*
631 *Signal Processing and Intelligent Systems (ICSPIS)*, page 1–6. IEEE.
- 632 Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial*
633 *Intelligence Research*, 11:169–198.
- 634 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N.,
635 Antiga, L., Desmaison, A., Kopf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S.,
636 Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance
637 deep learning library.
- 638 Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: Unbiased
639 boosting with categorical features. *Advances in Neural Information Processing Systems*, 31:6638–6648.
- 640 Robinson, J. M., Pasternak, Z., Mason, C. E., and Elhaik, E. (2021). Forensic applications of micro-
641 biomics: A review. *Frontiers in Microbiology*, Volume 11 - 2020.
- 642 Ruder, S. (2017). An overview of multi-task learning in deep neural networks.
- 643 Snyder, J. P. (1987). *Map Projections—A Working Manual*. U.S. Geological Survey Professional Paper
644 1395. U.S. Government Printing Office, Washington, DC.
- 645 Tang, L. (2024). Comparison the performances for distributed machine learning: Evidence from xgboost
646 and dnn. *Applied and Computational Engineering*, 103:209–215.
- 647 Zantvoort, K., Nacke, B., Görlich, D., Hornstein, S., Jacobi, C., and Funk, B. (2024). Estimation of
648 minimal data sets sizes for machine learning predictions in digital mental health interventions. *njp*
649 *Digital Medicine*, 7(1):361.
- 650 Zhang, Y., McCarthy, L., Ruff, E., and Elhaik, E. (2024). Microbiome geographic population structure
651 (mgps) detects fine-scale geography. *Genome Biology and Evolution*, 16(11):evae209.

652 **5. Supplementary Materials**

653 **5.1 Supplementary Tables**

654 **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

655 **Notes:** This table summarizes the architecture and training settings for each separate neural network used in the
 656 hierarchical pipeline. For each prediction stage (continent, city, coordinates), the hidden layers column specifies the
 657 number and size of fully connected layers, dropout indicates the range of dropout rates applied to reduce overfitting, batch
 658 normalization (Batch Norm) shows whether normalization was used, learning rate is the optimizer step size, batch size is
 659 the number of samples per training batch, and epochs is the total number of training iterations. These settings were
 660 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

661 **Notes:** Each row lists a parameter and its default value for the continent, city, and coordinate neural
 662 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

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

664 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}

665 **Notes:** This table details the architecture and training parameters for each branch of the Combined Neural Networks
666 model. Each branch (continent, city, coordinates) uses a specific set of hidden layers, dropout rates, and batch
667 normalization settings. The learning rate column indicates the optimizer step size for each branch. These parameters
668 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
<i>Architecture parameters</i>	
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
<i>Training parameters</i>	
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

669 **Notes:** Parameters are grouped by architecture and training settings, with their default values for the
670 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

⁶⁷¹ **Notes:** Each row lists a hyperparameter and the range or set of values explored during tuning for the
⁶⁷² combined neural network.

⁶⁷³ **5.1.3 GrowNet Parameters**

Table 9. Default parameters for hierarchical GrowNet model

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

⁶⁷⁴ **Notes:** Each row lists a parameter and its default value for the hierarchical GrowNet model, grouped
⁶⁷⁵ 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
<i>Hierarchical loss weights</i>	
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)

⁶⁷⁶ **Notes:** Each row lists a hyperparameter and the range or set of values explored during GrowNet
⁶⁷⁷ tuning, including hierarchical loss weights.

⁶⁷⁸ 5.1.4 Ensemble Learning Model Parameters

⁶⁷⁹ 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

⁶⁸⁰ **Notes:** Each row lists a parameter and its default value for XGBoost classification and regression
⁶⁸¹ 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

682 **Notes:** Each row lists a hyperparameter and the range or set of values explored during XGBoost
683 tuning.

684 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

685 **Notes:** Each row lists a parameter and its default value for LightGBM classification and regression
686 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

687 **Notes:** Each row lists a hyperparameter and the range or set of values explored during LightGBM
688 tuning.

689 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

690 **Notes:** Each row lists a parameter and its default value for CatBoost classification and regression
691 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)

692 **Notes:** Each row lists a hyperparameter and the range or set of values explored during CatBoost
693 tuning.

694 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

695 **Notes:** Each row lists a parameter and its default value for GrowNet classification and regression
696 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

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

699 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

700 **Notes:** Each row lists a parameter and its default value for neural network (MLP) classification and
701 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

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

704 5.1.10 TabPFN Parameters

Table 21. TabPFN model configuration

Parameter	Value
Model	Pre-trained TabPFN
Hyperparameter tuning	Max time

705 **Notes:** Each row lists a parameter or configuration for TabPFN models.

706 **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

707 **Notes:** This table describes the configuration parameters for the meta-models used in the ensemble learning pipeline.
 708 The meta-models (XGBoost classifiers) are trained on out-of-fold predictions from base models to optimally combine their
 709 outputs. Key parameters include the algorithm type, objective function, tree depth, learning rate, number of estimators,
 710 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

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

⁷¹⁵ 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

⁷¹⁶ **Notes:** Each row lists continent-level classification metrics (precision, recall, F1-score, support) for the
⁷¹⁷ separate neural network model.

⁷¹⁸ 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

⁷¹⁹ **Notes:** Each row lists continent-level classification metrics for the combined neural network model.

⁷²⁰ **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

⁷²¹ **Notes:** Each row lists continent-level classification metrics for the hierarchical GrowNet model.

⁷²² **5.1.15 Continent Classification: Ensemble Learning**

⁷²³ **Notes:** 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

⁷²⁴ **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

⁷²⁵ **Notes:** Each row lists city-level classification metrics for the separate neural network model.

⁷²⁶ **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

⁷²⁷ **Notes:** Each row lists city-level classification metrics for the combined neural network model.

⁷²⁸ **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

⁷²⁹ **Notes:** Each row lists city-level classification metrics for the hierarchical GrowNet model.

⁷³⁰ 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

⁷³¹ **Notes:** 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.

⁷³² **Notes:** Each row lists error group statistics for coordinate regression using the separate neural network
⁷³³ model.

⁷³⁴ 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

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

⁷³⁵ **Notes:** Each row lists error group statistics for coordinate regression using the combined neural
⁷³⁶ network model.

⁷³⁷ 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

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

⁷³⁸ **Notes:** Each row lists error group statistics for coordinate regression using the hierarchical GrowNet
⁷³⁹ model.

⁷⁴⁰ **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.

⁷⁴¹ **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

⁷⁴² **Notes:** Each row lists the percentage of predictions within specified radius thresholds for each model.

⁷⁴³ **5.2 Figures**

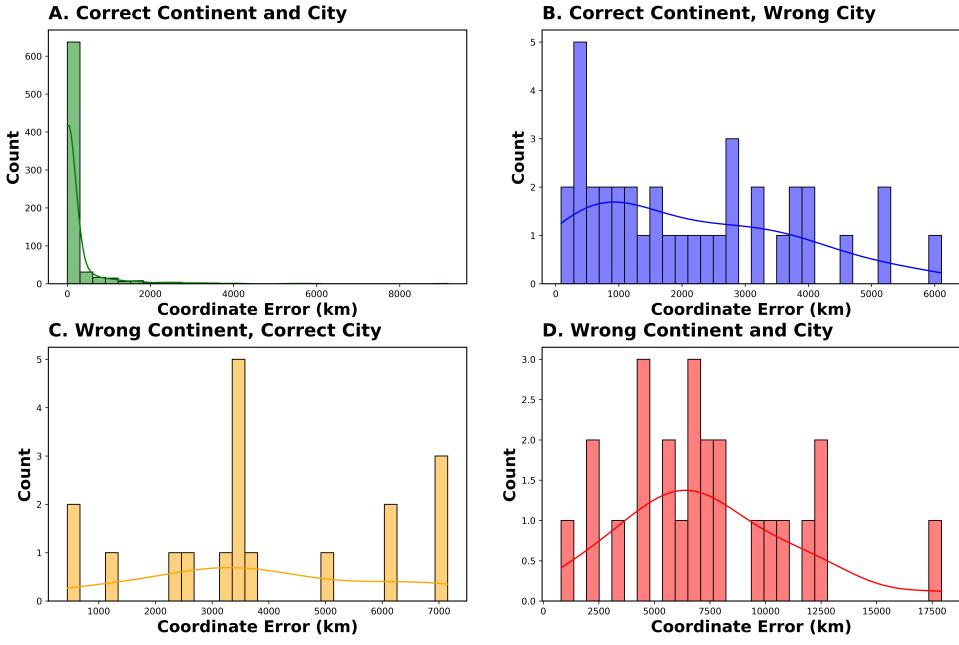


Figure 10. Error Distance Distribution by Continent and City for Ensemble Learning Model

Notes: 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.

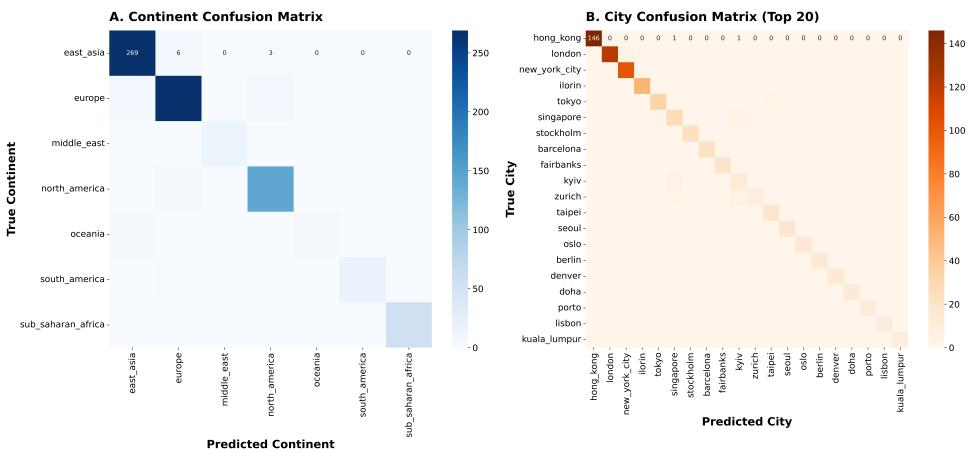


Figure 11. Confusion matrix for Continent and City Classification

Notes: 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.