Coordinates Are Just Features: Rethinking Spatial Dependence in Geospatial Modeling

Yameng Guo
Department of Business Informatics and Operations Management
Ghent University
Gent, Belgium
Seppe vanden Broucke
Department of Business Informatics and Operations Management
Ghent University
Gent, Belgium
Research Centre for Information Systems Engineering
KU Leuven
Leuven, Belgium
e-mail: {yameng.guo | seppe.vandenbroucke}@ugent.be

***Abstract*—Geospatial inference is crucial for various spatial prediction tasks, where the choice of modeling approach significantly impacts both inference performance and computational efficiency. Traditional geospatial statistical models, such as Geographically Weighted Regression (GWR) and Kriging, explicitly account for spatial dependence, but often come with high computational costs. In this study, we argue that treating coordinates as standard input features can yield competitive inference performance while significantly reducing computational costs when selecting a predictive model with an appropriate level of complexity. To support this, we compare geospatial statistical models with various machine learning approaches, including linear methods, tree ensemble methods, hybrid kernel-based methods that incorporate explicit geospatial learning, and a recent state-of-the-art tabular deep learning model—TabPFN—to assess their effectiveness in spatial prediction tasks (to the best of our knowledge, this is the first study to investigate the performance of TabPFN in the geospatial domain using explicit coordinate inputs). Our results demonstrate that when coordinates are sufficiently informative, tree-based ensemble models and tabular deep learning can implicitly capture spatial dependence without requiring explicit geospatial modeling, achieving superior performance whilst maintaining a reasonable computational cost.**

***Keywords-geospatial regression*; *ensembles modeling*; *spatial statistics*; *comparative performance*.**

I. Introduction

Spatial inference plays an increasingly critical role across various industries, including environmental science [1][2], urban planning [3], and disaster management [4][5], where predicting unobserved values at specific locations is essential.

Over the years, researchers have developed two primary approaches towards modeling spatial inference. *Explicit* approaches rely on the principle that geographically closer observations tend to be more similar. Traditional methods, such as Kriging [6][7] and Geographically Weighted Regression (GWR) [8], incorporate this principle through variograms or distance-decay weighting, offering both interpretability and predictive power which have been widely adopted for spatial interpolation and regression tasks.

Alternatively, Machine Learning (ML) models have emerged as powerful tools for handling large and complex datasets. These models treat coordinates as standard input features, allowing them to capture spatial dependence *implicitly*. Among them, tree-based ensembles—such as random forests and gradient boosting machines—excel at modeling nonlinear relationships and variable interactions. By incorporating spatial features into their predictive framework, they achieve competitive performance without the need for explicit geospatial modeling.

A hybrid approach has also gained traction, combining the interpretability of spatial models with the predictive power of machine learning. Techniques that integrate Kriging with ML-based kernels have demonstrated promising results by leveraging both domains' strengths [9][10].

The advancement of Tabular Deep Learning (TDL) further expands spatial inference possibilities. Whilst typically confronted with challenges, such as the need for extensive hyperparameter tuning and risk of overfitting, especially on small datasets, pre-trained models have appeared which aim to offer a robust alternative. For instance, the recently developed Prior-Data Fitted Network (PFN) Transformer [11], designed for tabular data, is pre-trained offline, enabling supervised learning on small datasets without additional hyperparameters tuning.

While traditional geospatial statistical models provide a rigorous framework for modeling geospatial dependence, they often struggle to balance predictive performance and computational efficiency, particularly with large datasets or nonlinear relationships. Conversely, TDL and ML—especially tree-based ensemble models—offer strong predictive performance with reasonable training and tuning costs as these models avoid constructing the geospatial distance function explicitly in a large scale.

Therefore, in this study, we reflect on the way traditional geospatial statistics leverage the distance matrix to model geospatial dependence and argue for the efficiency of considering the coordinates as just standard input features for spatial

inference tasks. We do so by presenting a comprehensive comparison of geospatial statistical models (e.g., Kriging and GWR), machine learning models (with a focus on tree ensembles), hybrid kernel-based models, and a state-of-the-art tabular deep learning model, i.e., TabPFN. Summarized, this work presents the following key contributions:

- We conduct a comparative experiment across statistical, ML, hybrid and TDL methods, with an emphasis on tree ensembles and TabPFN, to assess predictive performance and training efficiency;
- We analyze the practical considerations of training and tuning these models in real-world geospatial applications;
- We reflect on risks of putting a large emphasis on explicit spatial dependence usage, especially when coordinate information is sufficiently informative or strong ML models are available;
- The source code and datasets used in our work are publicly available on our GitHub page [12].

This paper is structured as follows: Section II provides a detailed explanation of related methodologies used in the field of geospatial reference. Section III introduces the experimental setup, covering the datasets, models, hyperparameter grids, and evaluation metrics used in the comparison. Section IV presents the results and discusses the effectiveness of all methods. The conclusion and future work are provided in Section V.

II. Methodology Review

This section clarifies the mechanisms underlying geospatial statistical models, ML, hybrid models and TabPFN, i.e., the techniques we will compare in this work, as well as their distinct ways to incorporate spatial dependence principles. By examining the mechanisms of these approaches, we aim to establish a foundation for comparing their performance and applicability in geospatial inference tasks.

A. Spatial Dependence-Based Models

Kriging and GWR are the most representative models in this group. Although they both rely heavily on the principle of spatial dependence, where observations close to each other are more similar than those farther apart, the emphasis of spatial relationships modeling of these two models are slightly varied.

1) Kriging: The main goal of Kriging is to quantify **spatial autocorrelation** to model and estimate the target values by using a variogram based on the assumption of jointly Gaussian distribution of the data, and then computes optimal weights for predictions by solving a system of linear equations, ensuring that predictions are best linear unbiased estimates.

The Kriging [13] predictor can be defined as:

$$\hat{Z}(\mathbf{s}_0) = \sum_{i=1}^{n} \lambda_i Z(\mathbf{s}_i),$$

where:

- $Z(\mathbf{s}_i)$: Observed value at location $\mathbf{s}_i$,
- λ_i: Weight assigned to $Z(\mathbf{s}_i)$, determined by spatial correlation.
- n: Number of observed locations.

The spatial correlation between locations is modeled using a **variogram** [14] which is defined as:

$$\gamma(h) = \frac{1}{2}\text{Var}[Z(\mathbf{s}) - Z(\mathbf{s}+\mathbf{h})],$$

where:

- h: Distance between two locations,
- $\gamma(h)$: Semi-variance at lag h.

By using the variogram, we can calculate the covariance matrix to solve the Kriging system,

$$C(\mathbf{s}_i, \mathbf{s}_j)\Lambda = C(\mathbf{s}_i, \mathbf{s}_0)$$

where Λ indicates the weight assigned to known nodes for the interpolation of an unknown node $\mathbf{s}_0$.

Based on the definition above, Kriging provides an estimate of prediction uncertainty that is defined as:

$$\sigma^2_{\text{Kriging}}(\mathbf{s}_0) = C(\mathbf{s}_0, \mathbf{s}_0) - \sum_{i=1}^{n} \lambda_i C(\mathbf{s}_i, \mathbf{s}_0) - \mu.$$

2) GWR: Compared with Kriging focusing on spatial autocorrelation and estimating the proximity similarity, GWR [15] is more based on the assumption of spatial heterogeneity. Though GWR also utilizes the distance matrix as weights to model the spatial variation, it fits a separate regression model locally at each location, weighting observations based on their proximity using a kernel function (e.g., Gaussian or bisquare), which allows for spatial variation in relationships between dependent and independent variables.

Essentially, the GWR can be defined as a linear combination:

$$y_i = \beta_0(\mathbf{s}_i) + \sum_{k=1}^{p} \beta_k(\mathbf{s}_i) x_{ki} + \epsilon_i,$$

where:

- y_i: Dependent variable at location $\mathbf{s}_i$,
- $\beta_0(\mathbf{s}_i)$ and $\beta_k(\mathbf{s}_i)$: Intercept and coefficient (for the k-th independent variable) at location $\mathbf{s}_i$,
- x_{ki}: Independent variable at location $\mathbf{s}_i$,
- ϵ_i: Random error term at location $\mathbf{s}_i$,
- p: Number of independent variables.

The regression coefficients $\boldsymbol{\beta}(\mathbf{s}_i)$ are estimated by solving the weighted least squares problem, which is expressed as

$$\boldsymbol{\beta}(\mathbf{s}_i) = \left(\mathbf{X}^\top \mathbf{W}(\mathbf{s}_i)\mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{W}(\mathbf{s}_i)\mathbf{y},$$

where $\mathbf{W}(\mathbf{s}_i)$ represents the diagonal weight matrix of the weights assigned to the location which is close to the point of interest.

To estimate the weight matrix, two kernel functions are commonly used, including:

- Gaussian kernel:

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{2b^2}\right),$$

- Bisquare kernel:

$$w_{ij} = \begin{cases} \left(1 - \left(\frac{d_{ij}}{b}\right)^2\right)^2 & \text{if } d_{ij} \leq b, \\ 0 & \text{if } d_{ij} > b, \end{cases}$$

where:

- d_{ij}: Distance between locations $\mathbf{s}_i$ and $\mathbf{s}_j$,
- b: Bandwidth parameter controlling the spatial extent of the weights.

Classical GWR models the local geospatial variation under the assumption of the same spatial scale, while a modification of GWR, namely Multiscale Geographically Weighted Regression (MGWR) [16], provides a more flexible and scalable framework by allowing different processes to operate at different spatial scales.

Although Kriging and GWR are widely used for spatial inference tasks, the application scenarios are slightly different. Kriging is more applied in spatial interpolation, such as estimating soil properties [17], pollutant concentrations [18][19], or precipitation levels [20], while GWR is more commonly applied in spatial regression scenarios, such as modeling house prices [21], socioeconomic factors [22], or environmental influences [23], where relationships vary spatially.

B. Machine Learning Models

Machine learning methods provide a data-driven approach to modeling, focusing on capturing patterns and relationships within the data without explicit assumptions about spatial dependence.

Typically, given a dataset $\{X, Y\}$ consisting of instances $\{x_i, y_i\}$ from a certain distribution $P(Y|X)$, the goal is to learn a function f that maps input features $\mathbf{x} \in \mathbb{R}^d$ to an output $y \in \mathbb{R}$. The general objective is:

$$\hat{f} = \arg\min_f \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i)),$$

where:

- $\ell(y_i, f(\mathbf{x}_i))$: Loss function measuring the error between predicted $f(\mathbf{x}_i)$ and actual y_i,
- n: Number of training instances.

To minimize the loss function (e.g., mean squared error for regression or cross-entropy for classification), a wide range of optimization algorithms, such as gradient descent and tree-based heuristics are developed to capture complex linear or nonlinear relationships between features. Specifically, tree ensemble models often outperform simpler models on structured data by building a series of decision trees and updating iteratively to minimize the loss,

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \gamma_m h_m(\mathbf{x}),$$

where:

- $f_m(\mathbf{x})$: Prediction at iteration m,
- $h_m(\mathbf{x})$: Weak learner (e.g., a shallow decision tree),
- γ_m: Step size for the weak learner.

Unlike the spatial dependence-based models which integrate the geospatial information explicitly, machine learning models theoretically are available for all kinds of tabular data inference tasks, but can be applied to the geospatial field easily by engineering the geographical features (e.g., raw coordinates, distance to landmarks, elevation, or land use types) and including location information (i.e., coordinates in most cases).

C. Hybrid Kernel-Based Models

Recent advances have sought to explore hybrid approaches to boost the strengths of handling of spatial dependence.

The most straightforward trail is to consider Kriging as an extension of GWR, but train these two components separately. Following this basic hybrid idea, Geographically Weighted Regression Kriging (GWRK) [24] was developed and its efficiency proven on datasets from different domains [25][26].

Another possible combination is merging Kriging with ML models. By using Kriging as the base model and ML models as either internal learners for residuals [27] or as a super learner [9], this hybrid approach helps mitigate the limitations of both model types, allowing effectively incorporating spatial relationships while enhancing predictive performance.

Moreover, the variogram function in Kriging or a local linear function are not the only choices to model geospatial dependence. E.g., Gaussian Processes (GPs) can also model spatial dependencies explicitly through kernel functions and by weighting proximal observations spatially. The Gaussian kernel is defined as:

$$k(\mathbf{s}_i, \mathbf{s}_j) = \exp\left(-\frac{\|\mathbf{s}_i - \mathbf{s}_j\|^2}{2\ell^2}\right),$$

where:

- $k(\mathbf{s}_i, \mathbf{s}_j)$: Covariance between points $\mathbf{s}_i$ and $\mathbf{s}_j$,
- ℓ: Length scale parameter, determining how quickly the correlation decays with distance,
- $\|\mathbf{s}_i - \mathbf{s}_j\|$: Euclidean distance between points $\mathbf{s}_i$ and $\mathbf{s}_j$.

In theory, by embedding spatial correlation into machine learning workflows, these kernel-based methods enhance predictive performance while retaining the capacity to model non-linearities and complex interactions.

D. TabPFN

TabPFN is a single Transformer pre-trained to approximate probabilistic inference using a designed prior based on Bayesian Neural Networks. It is built on Prior-Data Fitted Networks (PFNs) [28], which can directly sample from and approximate the Posterior Predictive Distribution (PPD). Unlike conventional neural networks and tree ensembles that rely on fixed structures, such as neural layers or constrained tree depth, TabPFN [11] incorporates not only a Bayesian Neural Network-based prior [29][30] but also Structural Causal Models [31][32] to capture complex feature dependencies and analyze underlying causal mechanisms, particularly in tabular data. It has demonstrated superior inference performance across various datasets spanning different domains. As a pre-trained Transformer, TabPFN embeds all input features as tokens and

processes them through a feed-forward mechanism, treating coordinates as standard input features alongside others.

In summary, these three types of models leverage geospatial dependence in two distinct ways: either by directly integrating geographic information as a distance matrix to model interactions between the target point and its proximal neighbors or by engineering proximity as hard features, incorporating location information as standard features while ignoring autocorrelation among points. Although a vast body of literature applies these methods to tackle various real-world challenges, researchers rarely discuss the advantages and efficiency of explicitly using spatial dependence. Models like Kriging and GWR often entail high computational costs and are susceptible to singular distance matrices, which can render the Kriging system or covariance matrix unsolvable.

In contrast, ML and TDL models mitigate computational cost and solvability concerns, as they do not require solving linear systems based on distance matrices. Instead, they directly model the mapping function from tabular features and approximate the prior distribution of the given dataset, which is particularly efficient with larger datasets.

To uncover the most efficient approach for different geospatial inference tasks, we conducted an extensive experiment evaluating various models in terms of predictive performance and computational cost. We hope this study provides new insights into modeling geospatial variables and selecting practical models for real-world applications, especially under the presence of stronger ML models, as well as very recent TDL approaches.

III. Experimental Setup

In this section, we describe an exhaustive experiment to compare a wide range of ML models with other well-known geospatial predictive modeling techniques, covering a collection of real-life datasets.

A. Datasets

There are two primary types of public datasets used in this work to evaluate the performance of geospatial statistical models and machine learning models, i.e., **property datasets** obtained from Kaggle and biology related datasets from the **R package Spatstat.data**.

All these datasets contain at least coordinates (either geographical or geometric coordinates), but not all of them have additional features, such as hedonic features of property data. To validate the capability of various models on capturing geospatial information and the utility of geospatial dependence, we divide the dataset further into two categories that consist of coordinates-only and full-feature datasets. Each dataset was cleaned to remove duplicate values and was re-scaled so features fall in a range of 0 to 1. We partitioned each dataset into a training set (70%), a validation set (10%) and a test set (20%). Note that when a timestamp was available (such as for real estate datasets), we perform the train-validation-test split in a temporal manner (i.e., chronologically).

Moreover, we carefully process the coordinates to ensure reliable geospatial inference. First, all coordinates are converted into a Cartesian coordinate system according to the dataset's geographical location, ensuring unified features to each model, and avoiding potential spherical distortions on statistic models which are based on distance matrices. Specifically, for GWR and Kriging, we keep the Cartesian coordinates unscaled to maintain consistent Euclidean distance calculations. For ML and TDL models, we scale the coordinates similarly to the other input features.

B. Models

As shown in Table I, we select a diverse set of models that cover different methodological categories to comprehensively evaluate the effectiveness of geospatial statistical models, ML and TDL approaches. The selected models are categorized into machine learning, TDL, kernel-based methods, and geospatial statistical models.

Machine learning models include Linear Regression with Ridge regularization [33], Support Vector Machine (SVM) [34][35][36]) and tree ensemble methods (Random Forest (RMF) [37], XGBoost [38], LightGBM (LGBM) [39], and CatBoost [40]). Strictly speaking, the kernel-based models covering Gaussian Processes [41], Tweedie Regression [42], and the hybrid Kriging-LGBM approach could also be placed under the ML group. But since they combine machine learning and geospatial statistics, we categorize them separately. The hybrid model—Kriging-LGBM [27], is the most representative in this group. It uses a LightGBM regressor as an internal kernel and then gathers and processes geospatial information with Kriging on target residuals. Moreover, a recent state-of-the-art tabular deep learning model—TabPFN—is also included in this experiment. To the best of our knowledge, this is the first study to investigate the performance of TabPFN in the geospatial domain using explicit coordinate inputs. Finally, we include the most classical geospatial statistical models, i.e., GWR [15], and Regression Kriging [43][44][45].

Each model's hyperparameters are tuned according to the grid values listed in the table, to ensure a fair and comprehensive evaluation across different modeling approaches. Hyperparameters for all models were systematically tuned on the validation set using root mean square error (RMSE). The best parameter combination was then used to test on the completely unseen test set to report the evaluation results. All models share the same data partitions. Note that TabPFN claims to be able to reach competitive results without any hyperparameter tuning, so for this pre-trained model, no tuning was performed.

C. Comparative Setup

To clarify the extent of importance of coordinates in various inference tasks and assess the efficiency of different models in terms of leveraging spatial locations, we evaluate the model performance under two main dataset configurations: one using only spatial coordinates, and the other one incorporating both coordinates and additional features when available.

TABLE I
OVERVIEW OF MODELS AND THEIR HYPERPARAMETERS USED IN THE COMPARISON.

Category	Type	Model	Hyperparameters
Machine Learning	Linear	Ridge LR	α: [0.1, 0.2, ..., 0.9]
		SVM	C: [1, 11, ..., 101]
			ϵ: [0.1, 0.2, ..., 0.9]
	Tree Ensemble	RandomForest	min_samples_split: [2, 3, 5]
			min_samples_leaf: [3, 5, 10]
		XGBoost	learning_rate: [0.1, 0.01, 0.005]
			reg_alpha: [0.0, 0.1, ..., 1.0]
			reg_lambda: [0.0, 0.1, ..., 1.0]
		LGBM	learning_rate: [0.1, 0.01, 0.005]
			reg_alpha: [0.0, 0.1, ..., 1.0]
			reg_lambda: [0.0, 0.1, ..., 1.0]
		CatBoost	iterations: [100, 200]
			learning_rate: [0.001, 0.005, 0.01, 0.05, 0.1]
			l2_leaf_reg: [0.1, 0.5, 1, 5]
Kernel Based	Gaussian	Gaussian Process	kernel: C(1.0) * RBF(length_scale_bounds=(1e-2, 1e2))
			alpha: [0.1, 0.2, ..., 0.9]
	Power	Tweedie	power: [0, 1, 1.2, 1.5, 1.8, 2, 3]
			alpha: [0.0, 0.1, ..., 0.9] + [2, 5, 8, 10]
	ML Kernel	Kriging LGBM	Kriging params: nlags = [30, 60, 90, 120]
			variogram_model: [“gaussian”, “linear”]
			Lightgbm params: reg_alpha: [0.0, 0.5, 1.0]
			reg_lambda: [0.0, 0.5, 1.0]
			learning_rate: [0.1, 0.01, 0.005]
Geospatial Statistics	Geospatial Heterogeneity	GWR	best bandwidth for kernel
	Geospatial Autocorrelation	Kriging	nlags: [30, 60, 90, 120]
			variogram_model: [“gaussian”, “linear”]
Deep Learning	Tabular DL	TabPFN	—

The primary evaluation metric to quantify the predictive performance of each model is the Root Mean Squared Error (RMSE). Additionally, we assess the computational efficiency by measuring the training time per model per run during the hyperparameter tuning. This dual assessment allows us to analyze the trade-offs between model performance and computational cost, providing insights into the practicality of each approach in geospatial prediction tasks.

The experiment is conducted on an Intel Core i9-13900 (13th Gen) CPU with 64 GB of RAM and an NVIDIA RTX A5000 GPU.

IV. DISCUSSION

All the results are shown in Table II and Table III. Since Regression Kriging only accepts coordinates as input, its predictive results remain the same for both datasets, with and without hedonic features.

Interestingly, we find that TabPFN consistently achieves the lowest RMSE across both datasets with hedonic features and without (coordinates only). In particular, on datasets with hedonic features, TabPFN outperforms all other models virtually all cases, which serves as one of the first illustrations of the competitive power of this recent approach in the domain of geospatial inference.

Tree ensemble models, especially LightGBM, XGBoost, and CatBoost, rank second. Although they do not surpass TabPFN, their performance is still clearly better than geospatial statistical models. In contrast, linear models, including Ridge Regression and SVM, consistently yield the worst predictions among all models, indicating that ML models can sufficiently capturing geospatial dependencies without having to deal with coordinates in a specific manner, as long as the chosen ML model is strong enough.

Notably, Gaussian Processes, Kriging, and the Kriging LGBM Regressor, which explicitly utilize geospatial information, also demonstrate strong performance on a few datasets where only coordinates were included. However, they do

TABLE II
COMPARISON OF MODEL PERFORMANCE (RMSE) ACROSS DIFFERENT DATASETS WITH **ONLY COORDINATE FEATURES**.

Data	Ridge LR	SVM	GWR	Kriging	Kriging LGBM	Gaussian P	Tweedie	RMF	LGBM	XGBoost	CatBoost	TabPFN
anemones	0.1756	0.1870	0.1841	0.1826	0.1826	0.1804	0.1755	0.1753	**0.1747**	0.1779	0.1766	0.1810
beijing	0.1833	0.1342	0.1390	0.1284	0.1284	0.1380	0.1833	0.1296	0.1279	0.1279	0.1273	**0.1272**
bronzefilter	0.1736	0.2364	0.2133	0.1835	0.1835	0.1754	0.1622	0.1553	0.1623	0.1615	0.1795	**0.1535**
dubai	0.1941	0.1584	0.1668	**0.1373**	**0.1373**	0.1539	0.1911	0.1384	0.1448	0.1413	0.1404	0.1391
london	0.0885	0.0717	0.0676	**0.0641**	**0.0641**	0.0704	0.0885	0.0643	0.0650	0.0652	0.0667	0.0653
longleaf	0.3114	0.2978	0.2546	0.2750	0.2750	0.2923	0.2531	0.2641	0.2798	0.2639	0.3037	**0.2451**
melbourne	0.0944	0.0708	0.0751	0.0603	0.0602	0.0652	0.0922	0.0608	0.0610	0.0599	0.0599	**0.0588**
newyork	0.1104	0.1018	0.0955	0.0964	0.0964	0.0981	0.1104	0.0928	0.0931	0.0930	0.0939	**0.0925**
paris	0.0216	0.0615	0.0208	0.0213	0.0213	0.0217	0.0216	0.0205	0.0203	0.0203	0.0203	**0.0202**
perth	0.0555	0.0444	0.0350	0.0348	0.0348	0.0384	0.0548	**0.0339**	0.0340	0.0341	0.0344	0.0340
seattle	0.1448	0.1181	0.1147	0.1101	0.1101	0.1154	0.1448	**0.1092**	0.1096	0.1095	0.1103	0.1100
spruces	0.2038	0.2361	0.1984	0.2284	0.2284	**0.1889**	0.1942	0.2204	**0.1889**	0.2004	0.1911	0.1928
waka	0.1240	0.1398	0.1237	0.1295	0.1295	0.1233	0.1235	0.1293	0.1235	0.1234	0.1232	**0.1230**

TABLE III
COMPARISON OF MODEL PERFORMANCE (RMSE) ACROSS DIFFERENT DATASETS WITH **COORDINATE AND ADDITIONAL FEATURES**.

Data	Ridge LR	SVM	GWR	Kriging	Kriging LGBM	Gaussian P	Tweedie	RMF	LGBM	XGBoost	CatBoost	TabPFN
beijing	0.1718	0.1378	0.1329	0.1284	0.1285	0.1608	0.1693	0.1031	**0.1003**	0.1045	0.1036	0.1008
dubai	0.1801	0.1982	0.1852	0.1373	0.1303	0.1982	0.1905	0.1194	0.1202	0.1201	0.1122	**0.1038**
london	0.0846	0.0757	0.0859	0.0641	0.0628	0.0776	0.0846	0.0589	0.0586	0.0588	0.0602	**0.0562**
melbourne	0.0803	0.0702	0.0673	0.0603	0.0389	0.0687	0.0581	0.0326	0.0296	0.0313	0.0291	**0.0263**
newyork	0.0863	0.0759	0.0721	0.0822	0.0726	0.0751	0.1002	0.0565	0.0562	0.0560	0.0561	**0.0532**
paris	0.0213	0.0246	0.0206	0.0213	0.0213	0.0217	0.0214	0.0202	0.0202	**0.0201**	**0.0201**	**0.0201**
perth	0.0494	0.0460	0.0355	0.0348	0.0324	0.0375	0.0489	**0.0270**	0.0277	0.0274	0.0282	0.0275
seattle	0.1252	0.1100	0.0966	0.1101	0.0981	0.1134	0.1253	0.0838	0.0820	0.0836	0.0831	**0.0790**

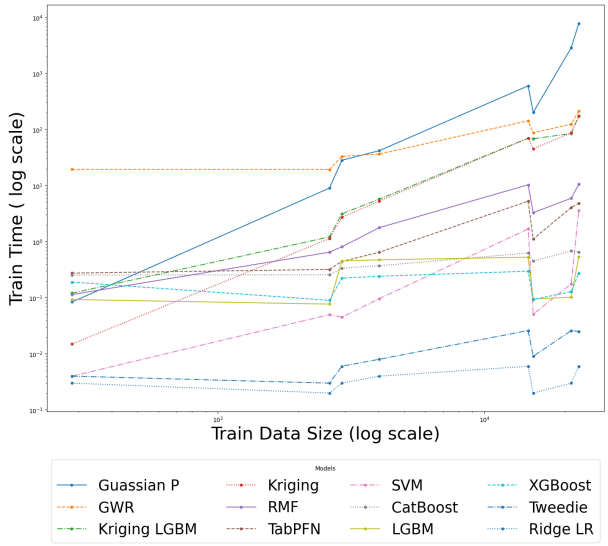

Figure 1. All features: visualizations of training time (s) per hyperparameter run across different models in log scale.

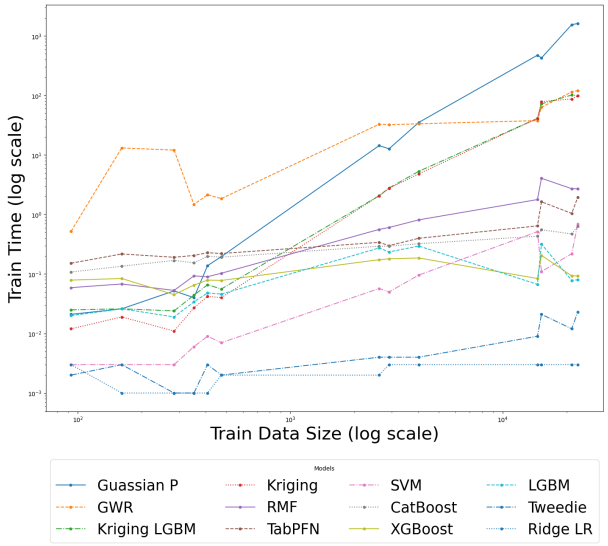

Figure 2. Coordinates features: visualizations of average training time (s) per hyperparameter run across different models in log scale.

encounter challenges in terms of incorporating additional features, limiting their effectiveness in such cases.

Figure 1 and Figure 2 evaluate model performance from a more practical perspective. Due to computational constraints, models that require less time for training and tuning are more advantageous for real-world applications. It is evident that models which are heavily reliant on spatial dependence (i.e., Gaussian Processes, Kriging, and GWR) experience exponentially increasing training and tuning times as the dataset size grows.

TabPFN, on the other hand, requires significantly less time due to its pre-trained nature. Given its superior predictive performance, TabPFN offers a balance between predictive power and efficiency. Similarly, tree ensemble models incur lower computational costs compared to statistical models, thanks to their optimized tree structures, which enable faster

training while maintaining competitive predictive performance.

Our experimental results highlight the utility of geospatial dependence in predictive modeling. Tabular deep learning, i.e., TabPFN and tree ensemble methods demonstrate strong predictive performance using only coordinates, as well as when additional features are included, in many cases outperforming traditional geospatial statistical models like GWR and Kriging. This suggests that explicit spatial modeling is not always necessary, especially when models are strong enough to implicitly capture spatial dependencies from coordinate features. Moreover, by treating coordinates as standard input features rather than relying on computationally intensive geospatial models, we can significantly reduce training and inference costs while maintaining competitive regression performance, which is particularly valuable for large-scale geospatial applications.

V. Conclusions

The primary goal of this work was to explore the balance between expressiveness, efficiency, and predictive power among different modeling approaches, including geospatial statistical models, machine learning models, kernel-based models, and tabular deep learning. Traditionally, geospatial inference research explicitly models spatial dependence by leveraging distance matrices. However, we argue that overemphasizing explicit spatial learning is not always necessary, as it neither guarantees superior predictive performance nor ensures computational efficiency compared to more effective approaches, such as tabular deep learning and tree ensemble models.

To further support our argument, we conducted a comparative experiment evaluating the predictive capabilities and computational costs of geospatial statistical models, machine learning models, kernel-based models, and tabular deep learning on datasets containing only coordinates, as well as datasets with additional features. The results demonstrate that TabPFN achieves an optimal balance between expressiveness, efficiency, and predictive power, making it the most effective choice for geospatial regression tasks in this study. These findings prompt a reconsideration of the learning paradigm in geospatial inference. Instead of relying on variograms or local functions based on distance metrics—which impose a heavy computational burden—incorporating coordinates as standard features in tabular deep learning or tree ensemble models may provide a more efficient and predictive alternative.

Although we have included several publicly available datasets, certain limitations should be acknowledged. A more exhaustive study should incorporate a wider range of datasets and modeling techniques from diverse fields beyond real estate, while also considering regions with varying population densities rather than focusing solely on highly urbanized areas. This would provide a broader and more generalizable evaluation. Additionally, future research could explore additional hybrid models, such as MGWR and GWRK, as well as expand the hyperparameter tuning grid to further optimize on performance.

References

[1] P. K. Rai, V. N. Mishra, and P. Singh, *Geospatial technology for landscape and environmental management: sustainable assessment and planning*. Springer, 2022.

[2] J. K. Thakur, S. K. Singh, A. Ramanathan, M. B. K. Prasad, and W. Gossel, *Geospatial techniques for managing environmental resources*. Springer Science & Business Media, 2012.

[3] B. Jiang and X. Yao, *Geospatial analysis and modelling of urban structure and dynamics*. Springer Science & Business Media, 2010, vol. 99.

[4] L. A. Manfré *et al.*, "An analysis of geospatial technologies for risk and natural disaster management", *ISPRS International Journal of Geo-Information*, vol. 1, no. 2, pp. 166–185, 2012.

[5] N. N. Kussul *et al.*, "Disaster risk assessment based on heterogeneous geospatial information", *Journal of Automation and Information Sciences*, vol. 42, no. 12, 2010.

[6] D. G. Krige, "A statistical approach to some basic mine valuation problems on the witwatersrand", *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 52, no. 6, pp. 119–139, 1951.

[7] G. Matheron, "Principles of geostatistics", *Economic geology*, vol. 58, no. 8, pp. 1246–1266, 1963.

[8] C. Bitter, G. F. Mulligan, and S. Dall'erba, "Incorporating spatial variation in housing attribute prices: A comparison of geographically weighted regression and the spatial expansion method", en, *Journal of Geographical Systems*, vol. 9, no. 1, pp. 7–27, Apr. 2007, ISSN: 1435-5949.

[9] G. Erdogan Erten, M. Yavuz, and C. V. Deutsch, "Combination of machine learning and kriging for spatial estimation of geological attributes", *Natural Resources Research*, vol. 31, no. 1, pp. 191–213, 2022.

[10] Z.-Y. Chen, R. Zhang, T.-H. Zhang, C.-Q. Ou, and Y. Guo, "A kriging-calibrated machine learning method for estimating daily ground-level no2 in mainland china", *Science of The Total Environment*, vol. 690, pp. 556–564, 2019.

[11] N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter, "Tabpfn: A transformer that solves small tabular classification problems in a second", *arXiv preprint arXiv:2207.01848*, 2022.

[12] ArmonGo, *Github: Geocoordsfeats*, Accessed: March 31, 2025.

[13] M. A. Oliver and R. Webster, "Basic steps in geostatistics: The variogram and kriging", Springer, Tech. Rep., 2015.

[14] M. Oliver and R. Webster, "A tutorial guide to geostatistics: Computing and modelling variograms and kriging", *Catena*, vol. 113, pp. 56–69, 2014.

[15] C. Brunsdon, S. Fotheringham, and M. Charlton, "Geographically weighted regression", *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 47, no. 3, pp. 431–443, 1998.

[16] A. S. Fotheringham, W. Yang, and W. Kang, "Multiscale geographically weighted regression (mgwr)", *Annals of the American Association of Geographers*, vol. 107, no. 6, pp. 1247–1265, 2017.

[17] Q. Zhu and H. Lin, "Comparing ordinary kriging and regression kriging for soil properties in contrasting landscapes", *Pedosphere*, vol. 20, no. 5, pp. 594–606, 2010.

[18] V. Van Zoest, F. B. Osei, G. Hoek, and A. Stein, "Spatio-temporal regression kriging for modelling urban no2 concentrations", *International journal of geographical information science*, vol. 34, no. 5, pp. 851–865, 2020.

[19] S. Araki, K. Yamamoto, and A. Kondo, "Application of regression kriging to air pollutant concentrations in japan with high spatial resolution", *Aerosol and Air Quality Research*, vol. 15, no. 1, pp. 234–241, 2015.

[20] M. P. Lucas *et al.*, "Optimizing automated kriging to improve spatial interpolation of monthly rainfall over complex terrain",

Journal of Hydrometeorology, vol. 23, no. 4, pp. 561–572, 2022.

[21] B. Huang, B. Wu, and M. Barry, "Geographically and temporally weighted regression for modeling spatio-temporal variation in house prices", *International journal of geographical information science*, vol. 24, no. 3, pp. 383–401, 2010.

[22] Z. Zhu, B. Li, Y. Zhao, Z. Zhao, and L. Chen, "Socio-economic impact mechanism of ecosystem services value, a pca-gwr approach", *Polish Journal of Environmental Studies*, vol. 30, no. 1, pp. 977–986, 2020.

[23] S. Li, Z. Zhao, X. Miaomiao, and Y. Wang, "Investigating spatial non-stationary and scale-dependent relationships between urban surface temperature and environmental factors using geographically weighted regression", *Environmental Modelling & Software*, vol. 25, no. 12, pp. 1789–1800, 2010.

[24] P. Harris, A. Fotheringham, R. Crespo, and M. Charlton, "The use of geographically weighted regression for spatial prediction: An evaluation of models using simulated data sets", *Mathematical Geosciences*, vol. 42, pp. 657–680, 2010.

[25] S. Kumar, R. Lal, and D. Liu, "A geographically weighted regression kriging approach for mapping soil organic carbon stock", *Geoderma*, vol. 189, pp. 627–634, 2012.

[26] M. Imran, A. Stein, and R. Zurita-Milla, "Using geographically weighted regression kriging for crop yield mapping in west africa", *International Journal of Geographical Information Science*, vol. 29, no. 2, pp. 234–257, 2015.

[27] B. S. Murphy, "Pykrige: Development of a kriging toolkit for python", in *AGU fall meeting abstracts*, vol. 2014, 2014, H51K–0753.

[28] S. Müller, N. Hollmann, S. P. Arango, J. Grabocka, and F. Hutter, "Transformers can do bayesian inference", *arXiv preprint arXiv:2112.10510*, 2021.

[29] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[30] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning", in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, ser. JMLR Workshop and Conference Proceedings, vol. 48, JMLR.org, 2016, pp. 1050–1059.

[31] J. Pearl, *Causality*. Cambridge university press, 2009.

[32] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[33] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems", *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970, ISSN: 0040-1706.

[34] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers", in *Proceedings of the fifth annual workshop on Computational learning theory*, ser. COLT '92, New York, NY, USA: Association for Computing Machinery, Jul. 1992, pp. 144–152, ISBN: 978-0-89791-497-0.

[35] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines", *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, Jul. 1998, ISSN: 2374-9423.

[36] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge: Cambridge University Press, 2000, ISBN: 978-0-521-78019-3.

[37] L. Breiman, "Random forests", English, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, ISSN: 0885-6125.

[38] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[39] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree", in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[40] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features", *Advances in neural information processing systems*, vol. 31, 2018.

[41] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[42] P. McCullagh, *Generalized linear models*. Routledge, 2019.

[43] I. O. Odeh, A. McBratney, and D. Chittleborough, "Further results on prediction of soil properties from terrain attributes: Heterotopic cokriging and regression-kriging", *Geoderma*, vol. 67, no. 3-4, pp. 215–226, 1995.

[44] T. Hengl, G. B. Heuvelink, and A. Stein, "A generic framework for spatial prediction of soil variables based on regression-kriging", *Geoderma*, vol. 120, no. 1-2, pp. 75–93, 2004.

[45] T. Hengl, G. B. Heuvelink, and D. G. Rossiter, "About regression-kriging: From equations to case studies", *Computers & geosciences*, vol. 33, no. 10, pp. 1301–1315, 2007.