Coordinates Are Just Features: A Benchmark Study of Geospatial Modeling

1st Yameng Guo
dept. Business Informatics and Operations Management
Ghent University
Gent, Belgium
0000-0003-2719-1356
yameng.guo@ugent.be

2nd Seppe vanden Broucke
dept. Business Informatics and Operations Management
Ghent University
Gent, Belgium
Research Centre for Information Systems Engineering
KU Leuven
Leuven, Belgium
0000-0002-8781-3906
seppe.vandenbroucke@ugent.be

***Abstract*—Geospatial inference has long been recognized as a critical topic of research. Modeling approaches in this area can generally be categorized into two main types, i.e., explicit and implicit spatial dependence learning. The key difference between these categories lies in whether spatial information (typically coordinates) is used as input to a distance function or simply treated as standard features in a machine learning algorithm. Traditional geospatial statistical models, such as Geographically Weighted Regression and Kriging, explicitly model spatial dependence. However, they often suffer from high computational costs and struggle to balance the trade-off between predictive performance and efficiency. In this work, we aim to demonstrate that explicitly modeling geospatial dependence is often not necessary. Treating coordinates as standard input features can yield competitive predictive performance while significantly reducing computational overhead, provided that a sufficiently capable learner is used. To substantiate our claims, we conduct an extensive comparison across a wide range of models. As an extended version of our previous work, we broaden the scope of models considered and include additional tabular deep learning models based on the transformer architecture and its attention mechanism. We also assess the statistical significance of performance differences across datasets. Furthermore, we include an interpretability analysis to examine the role of coordinates in models that learn spatial information either explicitly or implicitly. Our results show that even models, which treat coordinates as standard features can achieve competitive performance, with substantially lower training costs, while still effectively capturing spatial dependence. To the best of our knowledge, this is the first comprehensive study to evaluate both the effectiveness and efficiency of using coordinate inputs directly in spatial prediction tasks across a diverse set of modeling paradigms.**

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

I. Introduction

Spatial inference [1] plays a critical role across various industries, including environmental science [2] [3], urban planning [4], and disaster management [5] [6], where predicting unobserved values at specific locations is essential for informed decision-making in real-world applications.

To improve geospatial inference performance, researchers have developed numerous approaches that leverage both spatial and non-spatial information. Broadly, these approaches fall into two categories: explicit spatial dependence learning and implicit spatial dependence learning. Explicit spatial dependence learning relies on the principle that geographically closer observations are more likely to be similar. It typically treats location information (e.g., coordinates) as separate inputs and fits a distance function to estimate the influence of nearby points on a target location. Prominent methods such as Kriging [7] [8] and Geographically Weighted Regression (GWR) [9] embody this principle using variograms or distance-decay weighting, offering both interpretability and predictive power, and have been widely adopted for spatial interpolation and regression tasks.

In contrast, implicit spatial dependence learning does not explicitly estimate distance functions. Instead, it treats coordinates as regular features, integrating them into the model alongside other non-spatial features. This category includes traditional Machine Learning (ML) models such as linear regressors, Gaussian Processes (GPs), and tree ensembles, which excel at capturing nonlinear relationships while offering training efficiency.

Beyond classical ML, Tabular Deep Learning (TDL) has gained significant attention with the rise of deep learning. Transformers, in particular, provide powerful solutions for tabular tasks due to their flexibility and ability to model complex interactions among heterogeneous features. For example, the Prior-Data Fitted Network (PFN) Transformer [10] enables efficient supervised learning on small datasets without the need for additional hyperparameter tuning. Other deep learning models such as Neural Oblivious Decision Ensembles (NODE) [11] and Gated Additive Tree Ensemble (GATE) [12] have also shown promising performance in tabular settings.

In addition, there have been efforts to hybridize explicit and implicit approaches by developing models that combine geospatial distance functions with powerful ML learners, achieving promising results by leveraging the strengths of both paradigms [13] [14]. Despite the growing availability of modern approaches, however, traditional domains such as biology and agriculture still heavily rely on statistical geospatial models that explicitly learn spatial dependencies.

While these models offer strong interpretability and a theoretically sound framework, they often struggle with balancing predictive accuracy and computational efficiency, especially when dealing with large datasets or nonlinear patterns.

Conversely, ML and TDL are designed specifically for large datasets and non-linear distribution learning, which can offer better predictions more efficiently. Although we consider ML and TDL as competitive alternatives to explicit learning approaches, there are still some concerns regarding the hyperparameter tuning cost and potential overfitting issues of TDL. Moreover, due to the inherently spatially correlated nature of geospatial learning, the effectiveness of ML and TDL, which assume that instances are independent obviously raises questions. Therefore, to assess the feasibility of replacing explicit modeling with implicit learning in geospatial tasks and to investigate the effectiveness and efficiency across different model families, we conduct a comprehensive comparison of geospatial statistical models (e.g., Kriging and GWR), ML models (with a focus on tree ensembles), hybrid kernel-based models, and TDL model, e.g., TabPFN, NODE, GATE and etc.

In summary, this work makes the following key contributions:

- We conduct a comparative study across statistical, ML, hybrid, and TDL models to assess predictive performance and training efficiency, with a particular focus on how coordinates are used as input;
- We analyze the practical implications of training and tuning these models in real-world geospatial applications, using a wide range of datasets;
- We perform an exhaustive analysis that includes statistical significance testing and feature importance interpretation, shedding light on how explicit and implicit models utilize spatial information;
- To the best of our knowledge, this is the first benchmark study to systematically evaluate the use of geographic inputs across a broad range of models and datasets;
- All source code and datasets used in this study are publicly available on our GitHub page [15].

This paper is an extended version of our previous publication [1], presented at GEOProcessing 2025. The overall structure is as follows: Section II introduces the relevant methodologies used in geospatial modeling. Section III outlines the experimental setup, including datasets, models, hyperparameter configurations, evaluation metrics and significant test. Section IV presents and discusses the results, highlighting feature contributions in each model. Section V concludes the paper and outlines directions for future work.

II. Methodology Review

Prior benchmark studies have evaluated the performance of various models, ranging from classical ML approaches [16] to Deep Learning (DL) models [17] [18], across both real-world and synthetic datasets [19]. The most recent advancement at the time of writing, TabArena [20], has moreover introduced a dynamic benchmarking platform for tabular data that continuously integrates newly released datasets and models.

While these benchmarks provide exhaustive comparisons of model performance on tabular data, few studies examine how different types of specific feature inputs, particularly spatial features in this case, impact learning. In general-purpose tabular learning, this may not be a critical concern. However, in geospatial learning, where location information plays a central role in capturing spatial autocorrelation, the way spatial features are used becomes highly consequential. To address this gap, we conduct a benchmark study that focuses on the utility of coordinate-based features. Specifically, we compare the effectiveness and efficiency of explicit versus implicit spatial learning paradigms, laying a foundation for understanding their respective advantages in geospatial inference tasks.

This section presents an overview of the underlying mechanisms of the geospatial statistical models, machine learning approaches, hybrid models, and TDL methods evaluated in our work, with an emphasis on their distinct strategies for modeling spatial dependence.

A. Spatial Dependence-Based Models

Kriging and GWR are the most representative models in this group. Although they both heavily rely on the principle of spatial dependence, where observations close to each other are considered more similar than those farther apart, the emphasis of spatial relationships modeling of these two models is 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 a jointly Gaussian distribution of the data, followed by computing optimal weights for predictions by solving a system of linear equations, generating the linear unbiased estimates.

The Kriging [21] 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** [22], which is defined as:

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

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, which focuses on spatial autocorrelation and estimation of the proximity similarity, GWR [23] is 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, 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:

- 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) [24], provides a more flexible 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 typically applied in spatial interpolation, such as estimating soil properties [25], pollutant concentrations [26] [27], or precipitation levels [28], while GWR is commonly applied in spatial regression scenarios, such as modeling house prices [29], predicting socioeconomic factors [30], or environmental influences [31], 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 were 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 iteratively to minimize the overall 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, ML models are available for all kinds of tabular data inference tasks, but can be easily applied to the geospatial field by simply including location information (i.e., coordinates in most cases) as features, potentially with further feature-engineering efforts (e.g., distances to landmarks, elevation, land use types, aggregated census information, etc.).

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) [32] was developed and its efficiency proven on datasets from different domains [33] [34].

Another possible combination is to merge Kriging with ML models. By using Kriging as the base model and ML models as either internal learners for residuals [35] or as a super learner [13], 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. 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 ML workflows, these kernel-based methods enhance predictive performance while retaining the capacity to model non-linearities and complex interactions.

D. Tabular Deep Learning

Similar to traditional ML, TDL specifically targets tabular data using neural networks trained via backpropagation. A diverse range of model architectures have been developed to effectively handle the heterogeneous nature of tabular features, including Bayesian Neural Networks for uncertainty modeling, transformer-based architectures and attention mechanisms for capturing complex feature interactions, and ensemble-gated models that aim to combine the strengths of tree-based ensembles with neural networks through learned feature selection gates. We provide more detailed discussion on these approaches in the following sections.

1) TabPFN: TabPFN is a Transformer-based model that is pre-trained to carry out probabilistic inference under a carefully designed Bayesian-neural-network prior. Building on the PFN framework [36], it can draw direct samples from, and thus closely approximate, the posterior predictive distribution. Unlike classical neural networks or tree ensembles, whose expressiveness is limited by fixed layer counts or tree depth, TabPFN [10] enriches its prior with ideas from Bayesian neural networks [37], [38] and structural causal models [39], [40]. This combination lets it capture intricate feature dependencies and reason about causal relationships in tabular data. It's authors performed a thorough empirical study and showed that TabPFN delivers state-of-the-art inference accuracy. As a pre-trained Transformer, at inference time, TabPFN tokenises every input feature, including coordinates, and processes these tokens through the Transformer's feed-forward layers, yielding calibrated, sample-efficient predictions.

2) NODE: NODE [11] are inspired by gradient-boosted decision trees and use Oblivious Decision Trees (ODTs) as the fundamental building block. In ODTs, all nodes at a given depth split on the same feature and threshold, allowing for a structured and differentiable representation. Each ODT in the NODE architecture splits features based on a shared threshold at a given depth d, and the model outputs the sum of scaled leaf responses. Unlike traditional tree ensembles, NODE requires differentiability. Therefore, the discrete feature-splitting mechanism is replaced by a continuous variant, defined as:

$$f_i\hat{(x)} = \sum_{j=1}^{n} x_j \cdot entmax_\alpha(F_{ij}),$$

where:

- F: feature selection matrix,
- $entmax_\alpha$: α-entmax transformation [41].

The output of each layer is composed as a concatenation of the outputs of m individual trees $\left[\hat{h_1}, \hat{h_2}, \hat{h_3}, \ldots \hat{h_x}\right]$.

3) GATE: GATE [12] is a neural tree-based model that enhances the interpretability and performance of decision trees by incorporating neural network techniques such as gating and residual learning. Like NODE, it employs ODTs to learn nonlinear functions, but instead of averaging outputs, GATE uses Gated Feature Learning Units (GFLUs) to dynamically select and weight feature contributions.

The model output is defined as:

$$\hat{y} = \sigma\left(\sum_{i=1}^{M} \eta_i Y_i\right),$$

where:

- $\hat{y}$: final output,
- η: learnable parameters from gate mechanism,
- Y: the output of each sub-trees.

4) FT-Transformer: FT-Transformer [42] is another influential Transformer based model, tailored for tabular data. It combines a feature tokeniser with a transformer encoder, converting both categorical and numerical features into dense embeddings before applying multiple Transformer layers.

Given an input x, each feature x_j is first transformed to an embedding:

$$T_j = b_j + f_j(x_j) \in \mathbb{R}^d \qquad f_j : \mathbb{X}_\daleth \to \mathbb{R}^d$$

where:

- b_j: bias term,
- f_j: mapping function to map the feature to an embedding space.

After concatenating all embeddings for both categorical and numerical features, a series of Transformer layers F_i are applied sequentially:

$$T_i = F_i(T_{i-1}),$$

producing the final feature representation used for prediction.

5) DANets: DANets [43] build upon the FT-Transformer architecture by introducing instance-based attention. While maintaining the transformer backbone, DANets extend the attention mechanism by incorporating a learnable sparse mask in their abstract layer, enabling the model to capture complex and hierarchical feature interactions.

Given an input vector $f \in \mathbb{R}^m$ where m indicates feature numbers, a learnable sparse mask $M \in \mathbb{R}^m$ is used to filter the features by by element-wise multiplying with the vector f. The feature selection is defined as,

$$M = entmax_\alpha(W_{mask}), \qquad f' = M \odot f$$

where:

- $entmax_\alpha$: α-entmax transformation (same as NODE),
- W_{mask}: learnable vectors,
- M: a learnable sparse mask.

In conclusion, all models discussed above leverage geospatial dependence by either integrating spatial relationships through distance matrices, modeling interactions between a target point and its neighbors, or engineering spatial proximity as explicit input features, without directly modeling spatial autocorrelation.

While numerous real-world applications adopt these methods, the comparative efficiency and benefits of explicitly modeling spatial dependence remain underexplored. Traditional models such as Kriging and GWR rely heavily on spatial distance matrices, often leading to high computational costs and numerical issues, such as singular matrices, which makes them less scalable. In contrast, ML and TDL approaches avoid solving systems of equations derived from spatial relationships. Instead, they learn a direct mapping from feature space to the target variable, efficiently handling large datasets. However, they come with their own challenges, such as overfitting risks and increased computational burden from backpropagation.

To systematically assess the trade-offs, we conduct an extensive benchmark study evaluating each model's predictive performance and computational efficiency on geospatial inference tasks. Additionally, we apply feature attribution techniques to quantify the contribution of spatial features (e.g., coordinates), providing deeper insights into their role in model performance. We hope this study helps inform practical model selection for real-world geospatial applications, especially in light of increasingly powerful ML and TDL models.

III. Experimental Setup

In this section, we describe an exhaustive experiment to compare a wide range of ML models, statistics models, and TDL models, covering a collection of real-life datasets, with a complementary explanation of comparison hyperparameters, significance test and the interpretation approach.

A. Datasets

We utilize a vast collection of public datasets, primarily comprising real estate valuation datasets sourced from Kaggle, biology-related datasets from the R package `Spatstat.data`, and one additional well-known "yield" dataset [44].

TABLE I
Dataset summary with number of instances and features

Dataset	Nr. Instances	Nr. Features
singapore	9212	6
london	34994	10
melbourne	5759	12
newyork	4170	6
paris	21765	6
beijing	3745	13
perth	30210	9
seattle	20832	15
dubai	406	9
yield	1696	24
anemones	231	2
bronzefilter	678	2
longleaf	584	2
spruces	134	2
waka	504	2

A summary of the datasets used is provided in Table I, including the number of instances and features. The original features include both numerical and categorical variables, though all categorical features were transformed into a numerical format using the CatBoost encoder.

All datasets include spatial coordinates (either geographic or geometric) along with auxiliary features, such as hedonic attributes in the case of real estate valuation datasets. To investigate the impact of different spatial modeling strategies under different circumstances, we leverage each dataset to construct two variants: (i) coordinates-only, and (ii) all-features (original). Models are then trained on both variants to assess the effectiveness of explicit versus implicit spatial dependence modeling.

Prior to training, we first clean all datasets by converting all categorical features to encoded numerical ones (see remark above). We then continue by removing duplicates and rescaling features to the [0, 1] range to assure an equal playing ground. For those datasets containing timestamps, we then apply a temporal split (i.e., chronological partitioning) to construct training, validation, and test sets. For datasets without any temporal information, we perform a random split strategy. In either cases, we allocate 70% of the data for training, 10% for validation to be used for hyperparameter tuning, and 20% for testing and establish final performance metrics.

To ensure reliable geospatial inference, we carefully pre-process spatial coordinates as follows. All coordinates are converted into a Cartesian coordinate system, tailored to the geographic location of each dataset. This transformation standardizes the spatial input for all models and avoids distortions that may occur with spherical geometries, which is particularly relevant for statistical models relying on distance matrices. Specifically, for GWR and Kriging, these Cartesian coordinates remain unscaled to preserve accurate Euclidean distance computations whenever this distance metric is used, whilst for ML and TDL models, the coordinates are scaled in the same way as the other features. For the sake of notation, the unscaled Cartesian coordinates will be denoted as “lat” and “lon”, whereas the scaled ones will be labeled as “x” and “y” henceforth.

B. Models

We select a broad set of models for inclusion in our comparative study, spanning traditional ML, TDL, kernel-based methods, and geospatial statistical models. Serving as *statistical baselines*, Kriging [45] [46] [47] and GWR [23] are naturally included.

Alongside, we evaluate *machine learning* models, including the representative Linear Regression of Ridge regularization [48], and Support Vector Machines (SVM) [49] [50] [51]. In particular, we place special emphasis on tree ensemble methods, including Random Forest (RF) [52], XGBoost [53], LightGBM (LGBM) [54], and CatBoost [55]. These models have demonstrated consistently strong performance in practice, and in many benchmark studies on tabular data, they have even outperformed deep learning approaches [56].

To assess the benefit of *hybrid kernel-based architectures*, we include Kriging-LGBM [35], a two-stage model that uses a LightGBM regressor as the primary learner and applies Kriging to fit the residuals. Additionally, we include Gaussian and Power Tweedie. These two models are based on parametric assumptions about the target distribution (e.g., normal, Poisson-Gamma), while GPs explicitly incorporate spatial correlation using kernel functions that capture similarity between coordinate-based inputs.

The final group focuses on *TDL models*. As an extension of our previous work (where we focused solely on TabPFN), this study expands the scope to a much broader range of TDL architectures, including FT-Transformer, DANet, Gated Adaptive Network for Deep Automated Learning of Features (GANDALF) [57], GATE, and NODE.

To ensure a fair comparison across all models, we conduct an exhaustive hyperparameter tuning process. The hyperparameter grid used for all models is detailed in Table II. It is worth noting that not all models support or require extensive tuning. For instance, TabPFN, being a pre-trained model, is designed to achieve competitive results out-of-the-box by leveraging pre-trained weights without additional tuning. Moreover, other TDL models utilize an automatic learning rate optimization mechanism by default and are hence not listed separately.

C. Evaluation

All models are evaluated from two complementary perspectives: (i) predictive performance and (ii) computational cost (i.e., training time).

For predictive performance, we adopt Root Mean Square Error (RMSE) as the primary evaluation metric. Each model is initially trained on the training set, followed by systematic hyperparameter tuning on the validation set using RMSE as the selection criterion. The best-performing configuration is then used to evaluate the resulting final model’s predictive performance on the unseen test set. All models share identical data partitions and are assessed using the same consistent evaluation procedure.

To further investigate the role of spatial information in predictive performance, we evaluate each model under two distinct data configurations, as mentioned above, either using only spatial coordinates as features (“Coordinates-only”), or incorporating both coordinates and all additional attributes, when available (“All-features”). This comparison aims to highlight the importance of spatial information and to evaluate the models’ ability to capture spatial dependence, either explicitly or implicitly.

In terms of computational cost, we record the total training time required for each model across the entire hyperparameter tuning process, along with the number of tuning rounds. We then calculate the average training time per tuning round as the final comparison metric. This analysis provides insights into the computational efficiency of each method, enabling a comprehensive assessment of the trade-off between model accuracy and training overhead.

All the experiments are conducted on a standard workstation equipped with an Intel Core i9-13900 (13th Gen) CPU, 64 GB RAM, and an NVIDIA RTX A5000 GPU. All traditional machine learning models are trained using CPU resources, while the TDL models are trained with GPU acceleration.

D. Statistical Testing

To compare the performance of multiple algorithms across multiple datasets, we adopt the widely used Demšar method [58]. This approach is designed to assess whether the differences in performance among models are statistically significant by computing average ranks and visualizing them with a Critical Difference (CD) diagram.

Compared to parametric alternatives such as the paired t-test or ANOVA, the Demšar analysis is non-parametric, making no assumptions about data distribution and being especially suited for small-sample evaluations and providing robustness from its reliance on rank ordering rather than raw values. The method starts from ranking each model’s performance on every dataset, and then calculating average

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

Category	**Type**	**Model**	**Hyperparameters**
Geospatial Statistics	Geospatial Heterogeneity	GWR	best bandwidth for kernel
	Geospatial Autocorrelation	Kriging	nlags: [30, 60, 90, 120]
			variogram_model: [“gaussian”, “linear”]
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 parameters (same as base Kriging):
			nlags = [30, 60, 90, 120]
			variogram_model: [“gaussian”, “linear”]
			LGBM parameters:
			reg_alpha: [0.0, 0.5, 1.0]
			reg_lambda: [0.0, 0.5, 1.0]
			learning_rate: [0.1, 0.01, 0.005]
Deep Learning	Tabular DL	TabPFN	—
		FT-Transformer	num_heads: [4, 8, 16]
			attn_dropout: [0.0, 0.1, 0.2, 0.4]
		DANet	n_layers: [8, 20]; k: [3, 5, 8]
			dropout_rate: [0.1, 0.2, 0.3]
		GANDALF	gflu_stages: [2, 4, 6, 8, 10]
			gflu_dropout: [0, 0.1, 0.2, 0.3]
		GATE	gflu_stages: [2, 4, 6, 8, 10]
			gflu_dropout: [0, 0.1, 0.2, 0.3]
		NODE	num_layers: [1, 2, 4]
			num_trees: [8, 16, 32, 64]
			depth: [3, 4, 6]
			input_dropout: [0, 0.1, 0.2, 0.3]

ranks across all datasets. Next, a Friedman test is applied to assess whether there are any overall differences between the models. If such significance is detected, a post-hoc Nemenyi test is conducted to compare all pairs of models. Two models are considered significantly different if the difference in their average ranks exceeds a computed CD:

$$CD = q_\alpha \cdot \sqrt{\frac{m(m+1)}{6n}}$$

where:

- q_α: critical value from the Studentized range distribution in Nemenyi test,
- m: number of all models,
- n: number of all datasets,
- α: significance level.

We employ this framework to visualize outperforming models, with the detailed results presented in Section IV below.

E. Interpretation

To further investigate the role of individual features, particularly geographical coordinates, in geospatial learning, we conducted an interpretation analysis using SHAP (SHapley Additive exPlanations) values [59]. SHAP, grounded in game theory, provides a consistent method to quantify how much each feature contributes to a model's prediction by treating each feature as a player in a cooperative game and computing its marginal contribution across all feature subsets, which can be simply defined as a linear combination,

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i$$

where:

- $f(x)$: the prediction function,
- ϕ_0: average predictions of the model,
- M: number of all features,
- ϕ_i: shap values of each feature i.

And the Shapley values ϕ_i is defined as:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} (\hat{f}(S \cup \{i\}) - \hat{f}(S))$$

where:

- S: all possible subsets with all features except the i^{th} one,
- M: number of all features,
- $\hat{f}(S)$: predictions on subset.

IV. Results

In this section, we present our experimental results, encompassing both predictive performance and computational cost across multiple geospatial datasets. Additionally, we include a statistical model ranking analysis. A dedicated subsection is also provided for interpretability, using SHAP values to explore the contributions of different features to model predictions.

A. Performance & Cost

1) Performance: Table III reports the results across all datasets under the two configurations of the datasets: (i) coordinates and additional features (denoted as "Dataset (All)"), and (ii) only spatial coordinates (denoted as "Dataset (Coord)").

Among all models, TabPFN consistently achieves the lowest RMSE across both data configurations. Particularly for datasets with additional features, TabPFN outperforms all other models in nearly all cases, highlighting its strong generalization ability and effectiveness in handling geospatial inference tasks.

By comparison, other TDL architectures, namely FT-Transformer, DANet, GANDALF, and GATE, fall short of TabPFN's accuracy. Only NODE shows competitive behavior, outperforming the baseline on two coordinate-only settings. These findings imply that TDL models are not inherently superior, especially on the small, data-sparse problems typical of geospatial analysis. At the same time, TabPFN's recency highlights the considerable headroom that still exists for TDL methods in this setting.

Meanwhile, tree ensemble models, notably LightGBM, RMF, and CatBoost, exhibit consistently strong performance, frequently ranking first or second. Their competitive performance, even compared to TDL models, underscores the robustness of ensemble-based learners for tabular geospatial data. Indeed, this finding is in line with general prior studies [56] that conform the strength of these models, but highlights in our context that they are well capable to treat coordinates as any other feature.

In contrast, linear models, such as Ridge Regression, generally perform worse than statistical geospatial models like Kriging and GWR. This further confirms the limitation of simple linear assumptions in order to capture spatial heterogeneity.

The statistical models, GWR and Kriging, however, do not outperform TabPFN or ensemble models overall but do demonstrate notable effectiveness in the coordinates-only setting. This aligns with their theoretical strengths in capturing explicit spatial autocorrelation, particularly when supplementary features are unavailable.

In summary, TabPFN offers state-of-the-art performance across both feature configurations, proving its capacity to implicitly learn spatial dependencies. This finding is in line with our previous work [1], but our exhaustive study performed here shows that other TDL approaches do not achieve the same result. Next, tree ensemble models remain highly effective and computationally efficient alternatives, especially when training data is limited. Finally, statistical models (GWR and Kriging) do retain their relevance in coordinate-only scenarios, validating their role in explicitly modeling spatial structure.

More importantly, it is notable that the comparison between datasets with and without additional features confirms the importance of complementary information. We see that across

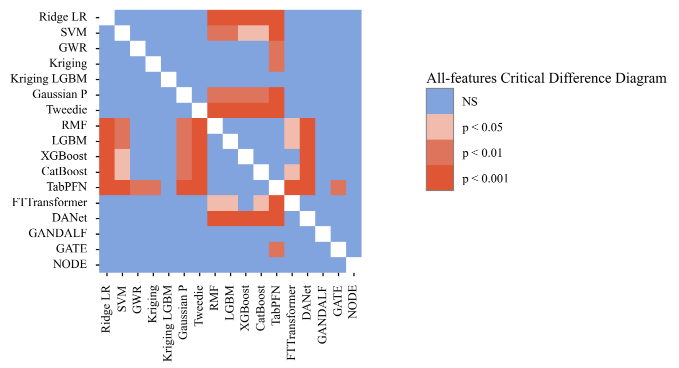

Figure 1. All features: visualizations of Demšar analysis on performance significance.

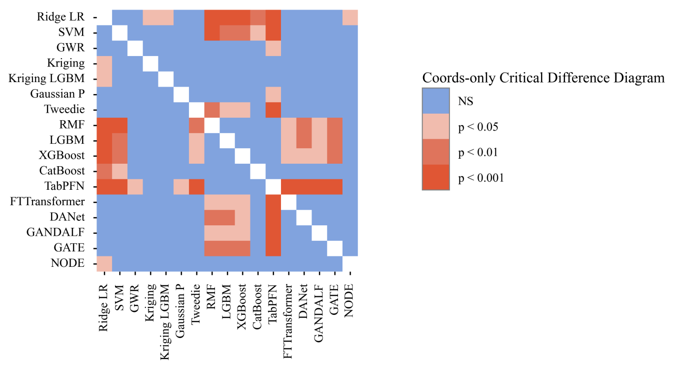

Figure 2. Coordinate features: visualizations of Demšar analysis on performance significance.

nearly all models, the inclusion of additional attributes leads to substantially lower RMSE, emphasizing the necessity of domain-specific features being available, and the benefit of rich feature engineering in geospatial learning.

2) Significance: To support our comparison results, we conduct a statistical significance analysis using the method proposed by Demšar [58]. First, the average rank of each model was computed across all datasets. After confirming the significance of a Friedman test, We then applied a Nemenyi's post-hoc test for pairwise comparisons, identifying statistically significant differences between models based on whether the difference in their average ranks exceeded the CD threshold.

The results are visualized in Figure 1 and Figure 2, where the X and Y-axes denote the model names. Each grid cell is color-coded to reflect the level of statistical significance between each model pair.

When using datasets with both coordinates and additional features, TabPFN significantly outperforms all other models, including both statistical models (GWR and Kriging) and other deep learning techniques. Whereas tree ensemble models (e.g., LightGBM, Random Forest, and CatBoost) do not achieve statistically significant improvements over GWR or Kriging, they do significantly outperform weaker models such as Gaussian Process and Tweedie Regressor, both of which explicitly encode spatial distance in their modeling.

For datasets with only spatial coordinates, fewer statistically significant differences are observed across models. Even the best-performing model, TabPFN, is not significantly better than Kriging, though it still significantly outperforms GWR. This result reinforces the value of explicit spatial modeling techniques (e.g., Kriging) when no additional features are available.

We also present two CD diagrams in Figure 3 (for datasets with all features) and Figure 4 (for coordinates-only datasets), which illustrate the average rank of each model across all datasets. As a refresher: models are ordered from best (left) to worst (right) along the X-axis. Each model is marked with a star to indicate its rank. Horizontal bars are used to group models that do not differ significantly from each other.

In the all-feature setting, TabPFN achieves the highest average rank and is significantly better than statistical models. The family of tree ensemble models follow closely and, although not statistically superior to TabPFN, are substantially better than linear models, most TDL models and statistical models, which occupy the lowest ranks.

In the coordinates-only setting, TabPFN again ranks first but is not significantly better than Kriging, corroborating earlier findings that explicit spatial models remain competitive when limited to spatial coordinates alone.

3) Computational Cost: We assess computational cost in terms of training time, as shown in Figure 5 and Figure 6. Training efficiency is a critical factor for the deployment of geospatial models in practice, especially in large-scale or resource-constrained environments.

Traditional geospatial statistical models, such as GPs, Kriging, and GWR, incur significantly higher computational costs, with training time increasing exponentially as dataset size grows. These models consistently record the longest training durations across all experiments, primarily due to their reliance on spatial dependence structures and costly matrix operations.

In contrast, TabPFN achieves high efficiency owing to its pre-trained foundation model, requiring reasonable training and tuning time when used on a new dataset. This, combined with its superior predictive accuracy, positions TabPFN as a highly effective model that balances both performance and efficiency.

While other TDL models (e.g., transformer-based or attention-based architectures) benefit from GPU acceleration, they still exhibit considerably higher training costs compared to traditional machine learning models. Despite this, TDL models generally outperform statistical baselines when additional features are included.

Among traditional machine learning models, tree ensemble methods (e.g., LightGBM, Random Forest, CatBoost) demonstrate a favorable trade-off between computational cost and predictive accuracy. These models benefit from optimized implementations and efficient tree construction algorithms, leading to relatively low training times even on larger datasets.

Overall, our results suggest that explicit spatial modeling is not always necessary, particularly when strong predictive

TABLE III
COMPARISON OF MODEL PERFORMANCE (RMSE) ACROSS DIFFERENT DATASETS. RESULTS ARE GROUPED BY WHETHER ADDITIONAL FEATURES ARE USED OR ONLY COORDINATES. BEST SCORES ARE IN **BOLD**, SECOND-BEST IN UNDERLINE.

Dataset (All)	**Ridge LR**	**SVM**	**GWR**	**Kriging**	**Kriging LGBM**	**Gaussian P**	**Tweedie**	**RMF**	**LGBM**	**XGBoost**	**CatBoost**	**TabPFN**	**FTTransformer**	**DANet**	**GANDALF**	**GATE**	**NODE**
beijing	0.1718	0.1393	0.1335	0.1284	0.1285	0.1608	0.1693	0.1034	0.1003	0.1045	0.1050	0.1009	0.1403	0.3104	**0.0959**	0.1659	0.1000
dubai	0.1801	0.1981	0.1852	0.1373	0.1303	0.1982	0.1905	0.1218	0.1202	0.1201	0.1112	**0.1042**	0.1776	0.2127	0.1850	0.1899	0.1377
london	0.0824	0.0717	0.0659	0.0637	0.0621	0.0732	0.0824	0.0577	0.0574	0.0583	0.0595	**0.0557**	0.0666	0.0827	0.0655	0.0646	0.0622
melbourne	0.0803	0.0702	0.0673	0.0603	0.0389	0.0687	0.0581	0.0327	0.0296	0.0313	0.0289	**0.0263**	0.0592	0.0944	0.0472	0.0435	0.0391
newyork	0.0868	0.0742	0.0858	0.0816	0.0684	0.0726	0.0883	0.0575	0.0578	0.0581	0.0584	**0.0567**	0.0736	0.0797	0.0694	0.0823	0.0634
paris	0.0213	0.0246	0.0206	0.0213	0.0213	0.0217	0.0214	**0.0201**	0.0202	0.0203	0.0202	**0.0201**	0.0216	0.0625	0.0207	0.0214	0.0207
perth	0.0494	0.0459	0.0355	0.0348	0.0324	0.0375	0.0489	**0.0269**	0.0277	0.0275	0.0279	0.0274	0.0383	0.0440	0.0320	0.0337	0.0306
seattle	0.1252	0.1100	0.0967	0.1101	0.0981	0.1134	0.1253	0.0833	0.0820	0.0830	0.0834	**0.0791**	0.0918	0.1034	0.0894	0.0871	0.0872
singapore	0.1057	0.0715	0.0654	0.1090	0.0780	0.0803	0.0975	0.0574	0.0563	0.0575	0.0547	**0.0510**	0.0749	0.0834	0.0722	0.0744	0.0677
yield	0.1314	0.0778	0.1141	0.0554	0.0580	0.0967	0.1278	0.0577	0.0566	0.0550	0.0541	**0.0498**	0.1541	0.1137	0.1349	0.1288	0.1198
Dataset (Coord)	**Ridge LR**	**SVM**	**GWR**	**Kriging**	**Kriging LGBM**	**Gaussian P**	**Tweedie**	**RMF**	**LGBM**	**XGBoost**	**CatBoost**	**TabPFN**	**FTTransformer**	**DANet**	**GANDALF**	**GATE**	**NODE**
beijing	0.1833	0.1342	0.1377	0.1284	0.1284	0.1380	0.1833	0.1294	0.1279	0.1279	0.1279	0.1272	0.1318	0.1304	0.1341	0.1815	**0.1261**
dubai	0.1941	0.1583	0.1692	**0.1373**	**0.1373**	0.1539	0.1911	0.1376	0.1448	0.1413	0.1405	0.1390	0.1701	0.1896	0.1859	0.1935	0.1580
london	0.0858	0.0702	0.0656	0.0637	0.0637	0.0690	0.0858	**0.0627**	0.0640	0.0643	0.0657	0.0641	0.0708	0.0685	0.0699	0.0843	0.0670
melbourne	0.0944	0.0708	0.0758	0.0602	0.0603	0.0652	0.0922	0.0605	0.0610	0.0599	0.0604	**0.0588**	0.0664	0.0945	0.0786	0.0812	0.0621
newyork	0.1060	0.0973	0.0921	0.0908	0.0908	0.0919	0.1062	0.0879	0.0882	0.0886	0.0890	**0.0876**	0.0961	0.0914	0.0928	0.1080	0.0916
paris	0.0216	0.0614	0.0205	0.0213	0.0213	0.0217	0.0216	0.0205	0.0203	0.0203	0.0203	**0.0202**	0.0215	0.0215	0.0208	0.0212	0.0203
perth	0.0555	0.0444	0.0349	0.0348	0.0348	0.0384	0.0548	**0.0339**	0.0340	0.0341	0.0343	0.0340	0.0376	0.0379	0.0366	0.0473	0.0356
seattle	0.1448	0.1181	0.1141	0.1101	0.1101	0.1154	0.1448	**0.1089**	0.1096	0.1096	0.1102	0.1100	0.1190	0.1135	0.1186	0.1236	0.1143
singapore	0.1524	0.1328	0.1334	0.1090	0.1090	0.1401	0.1524	**0.1059**	0.1171	0.1152	0.1207	0.1096	0.1342	0.1361	0.1354	0.1406	0.1333
yield	0.2379	0.0674	0.0805	0.0517	0.0517	0.0732	0.2277	0.0535	0.0549	0.0545	0.0529	**0.0487**	0.1335	0.0876	0.0781	0.2715	0.0685
anemones	0.1756	0.1869	0.1856	0.1826	0.1826	0.1804	0.1755	0.1755	**0.1747**	0.1779	0.1768	0.1808	0.1817	0.2238	0.2126	0.1855	0.1921
bronzefilter	0.1736	0.2364	0.2119	0.1835	0.1835	0.1754	0.1622	0.1555	0.1623	0.1615	0.1814	**0.1535**	0.2849	0.1778	0.1929	0.2635	0.1729
longleaf	0.3114	0.2978	0.2545	0.2750	0.2750	0.2923	0.2531	0.2608	0.2798	0.2639	0.3036	**0.2460**	0.2564	0.2673	0.2573	0.2469	0.2524
spruces	0.2038	0.2361	0.1972	0.2284	0.2284	0.1889	0.1942	0.2167	0.1889	0.2004	0.1911	0.1935	0.1989	0.2167	0.1992	0.1874	**0.1861**
waka	0.1240	0.1398	0.1237	0.1295	0.1295	0.1233	0.1235	0.1282	0.1235	0.1234	0.1237	**0.1232**	0.1319	0.1442	0.1283	0.1236	0.1283

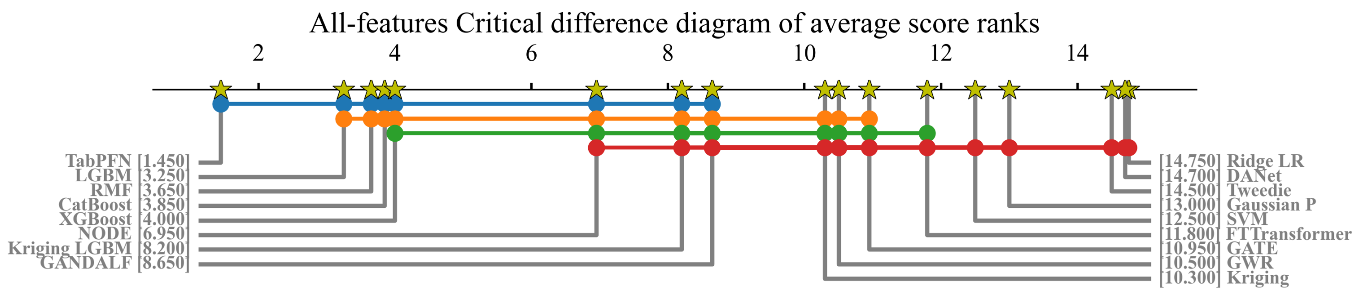

Figure 3. All features: visualizations of Demšar analysis on performance ranking.

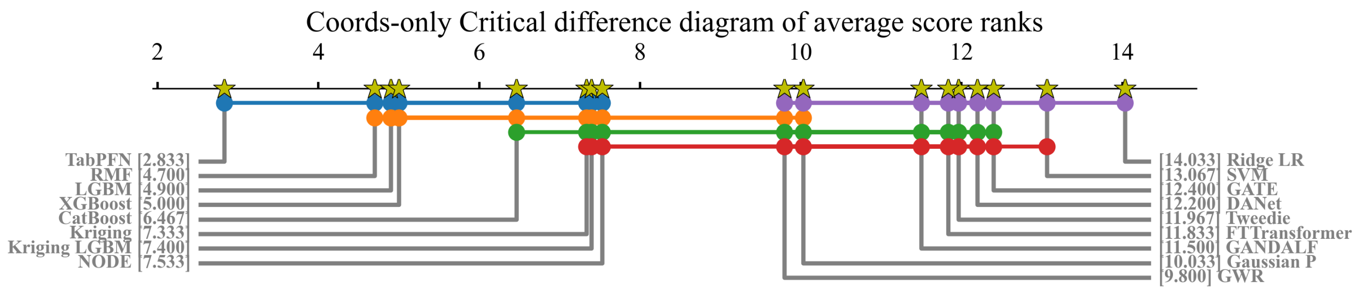

Figure 4. Coordinate features: visualizations of Demšar analysis on performance ranking.

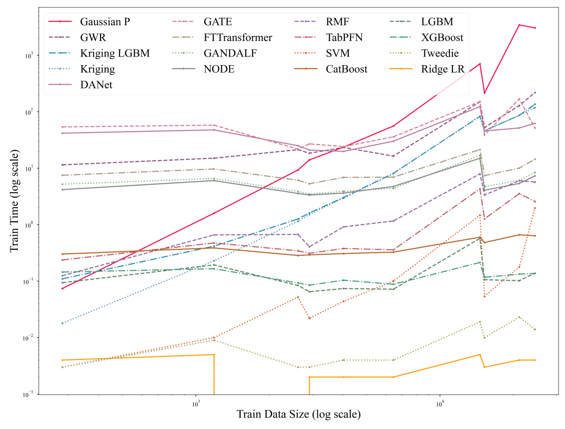

Figure 5. All features: visualizations of training time (s) per hyperparameter run across different models in log scale.

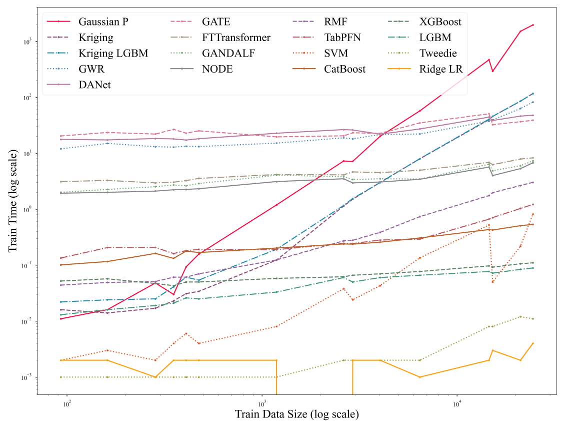

Figure 6. Coordinate features: visualizations of average training time (s) per hyperparameter run across different models in log scale.

models and additional contextual features are available, as is often the case in real-life settings. Tabular learners, including TabPFN and tree ensembles, are capable of implicitly capturing spatial dependencies from coordinate inputs alone. By treating spatial coordinates as standard input features rather than modeling geospatial dependencies explicitly, one can significantly reduce computational overhead while preserving or even improving predictive accuracy. This efficiency advantage is especially valuable for scalable geospatial applications where frequent retraining or rapid deployment is required.

B. Interpretation

Due to the architectural diversity across models, we adopted model-specific SHAP approximations. For deep learning models, we applied Gradient SHAP [59], which is tailored to differentiable models and estimates SHAP values by computing the expectation of gradients relative to randomly sampled baselines. For the other models, including tree ensembles, linear models, and statistical baselines, we employed the Kernel SHAP explainer [59], a model-agnostic method that perturbs inputs and fits a locally weighted linear model to

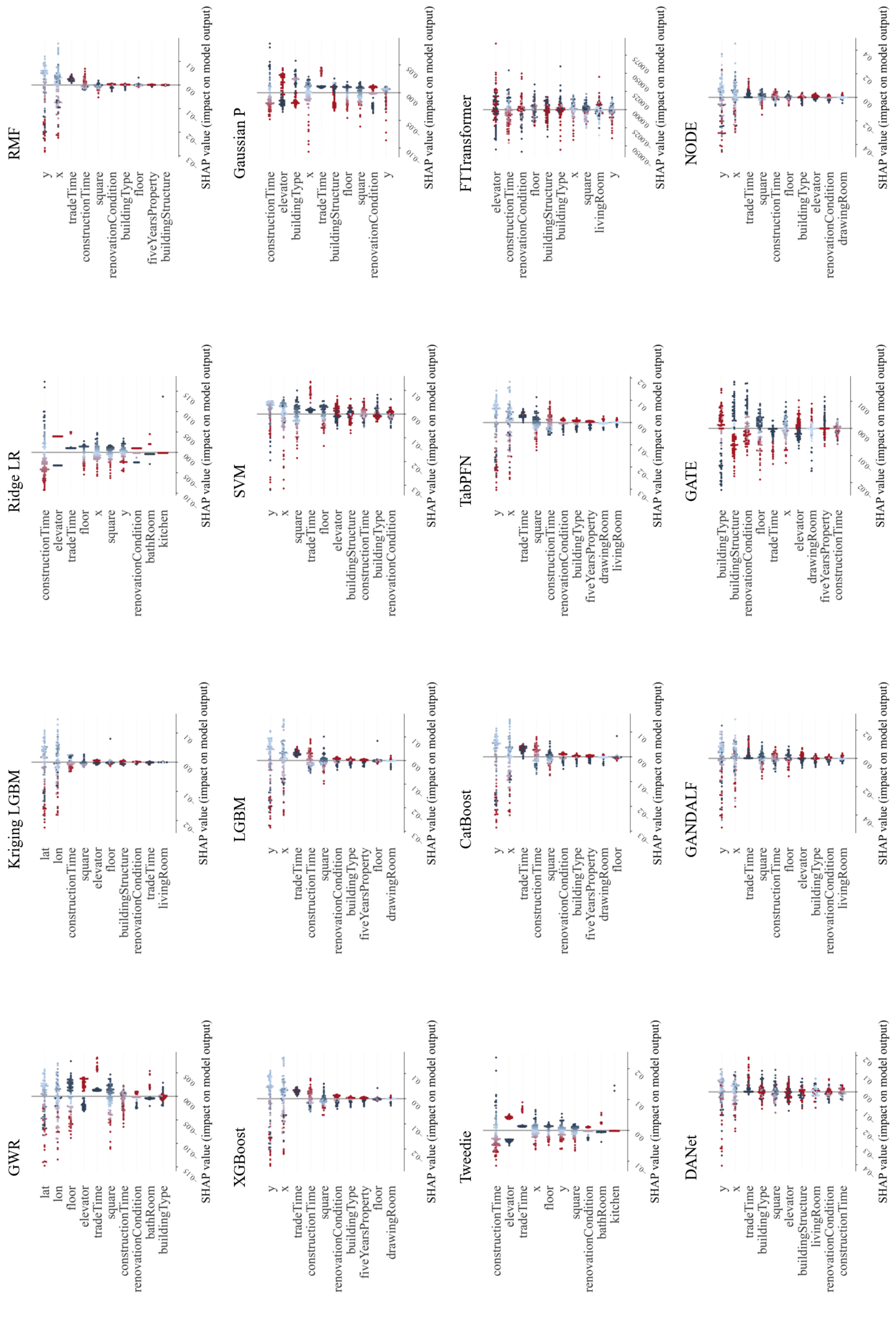

Figure 7. Feature attributions using the Beijing property dataset.

approximate each feature's contribution.

To elaborate this analysis, we select the "Beijing" property dataset and randomly sample 1000 instances as a representative test set. Feature attributions were then computed for all models and ranked based on their importance scores. For models explicitly leveraging geospatial dependencies (e.g., GWR, Kriging), spatial coordinates are labeled as "lat" and "lon". For models treating coordinates as standard tabular features, the coordinate inputs are denoted "x" and "y".

As shown in Figure 7, most models effectively utilize spatial information, either implicitly or explicitly. Noteworthy is that models with weaker predictive performance, such as NODE and Ridge Regression, assign lower importance to coordinate features, indicating their limited ability to extract spatial patterns. In contrast, strong-performing models, such as TabPFN, tree ensembles, and Kriging, consistently rank spatial features amongst the most influential.

These findings reinforce the notion that explicit spatial modeling is not strictly necessary for successful geospatial inference. Modern tabular learners, when sufficiently expressive, can implicitly learn geospatial patterns from coordinate features without specialized geospatial mechanisms. This highlights a promising alternative to traditional geo-statistical modeling, particularly when working with feature-rich tabular datasets.

V. Conclusions

The primary goal of this work was to explore the distinction between explicit and implicit spatial learning in the context of geospatial inference. Traditionally, statistical models such as Kriging and GWR capture spatial autocorrelation explicitly through distance-based functions. However, instead of relying on these computationally intensive distance-based modeling approaches, an alternative is simply to treat coordinates as standard input features and leverage them within ML or TDL models, given that they are strong enough. This implicit strategy can yield competitive predictive performance while significantly reducing computational costs.

To support this argument, we conducted a comprehensive evaluation across a wide range of models for geospatial regression tasks, including traditional statistical approaches, linear and ensemble-based ML methods, and recent TDL architectures. Our benchmark considered both datasets with only coordinate features and those augmented with additional geospatial attributes. The results reveal clear distinctions in predictive accuracy, training efficiency, and interpretability across these different model families.

Our findings indicate that whilst geo-statistical models do remain relevant for specific coordinate-only settings, general-purpose tabular learners, particularly TabPFN and ensemble tree models, emerge as powerful, scalable, and interpretable alternatives for modern geospatial learning. Furthermore, the SHAP-based feature attribution analysis demonstrates that top-performing models such as TabPFN and tree ensembles are capable of leveraging spatial features effectively, and do so without explicit spatial modeling. In contrast, weaker models tend to underutilize spatial information, reinforcing the importance of model capacity in capturing geospatial dependencies.

In summary, our results call for a re-evaluation of the traditional spatial learning paradigm. They demonstrate the feasibility and efficiency of treating location information as standard input features, empowering both ML and TDL models to perform robust geospatial inference in real-world applications.

We strongly extended the scope of models and datasets compared to our previous research. Nevertheless, further work is needed and invited to solidify these findings. Due to the limited availability of large-scale public datasets, we were unable to fully assess TDL models in extreme data-rich settings, where deep learning typically thrives. Additionally, most of our evaluations were conducted in highly urbanized areas, offering limited insight into how these learning paradigms perform in sparsely populated or rural regions. Future research could also explore more hybrid models, or zoom into upcoming transformer-based architectures. Finally, a best-effort fair choice was made in this study in terms of hyperparameter tuning, which could be optimized further across all models to potentially enhance top-performance levels.

References

[1] Y. Guo and S. vanden Broucke, "Coordinates are just features: Rethinking spatial dependence in geospatial modeling," in *GEOProcessing 2025, The Seventeenth International Conference on Advanced Geographic Information Systems, Applications, and Services*, pp. 48–55.

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

[3] 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.

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

[5] L. A. Manfré, E. Hirata, J. B. Silva, E. J. Shinohara, M. A. Giannotti, A. P. C. Larocca, and J. A. Quintanilha, "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.

[6] N. N. Kussul, B. V. Sokolov, Y. I. Zyelyk, V. A. Zelentsov, S. V. Skakun, and A. Y. Shelestov, "Disaster risk assessment based on heterogeneous geospatial information," *Journal of Automation and Information Sciences*, vol. 42, no. 12, 2010.

[7] 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.

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

[9] 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," *Journal of Geographical Systems*, vol. 9, no. 1, pp. 7–27, Apr. 2007.

[10] 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.

[11] S. Popov, S. Morozov, and A. Babenko, "Neural oblivious decision ensembles for deep learning on tabular data," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[12] M. Joseph and H. Raj, "GATE: gated additive tree ensemble for tabular classification and regression," *CoRR*, vol. abs/2207.08548, 2022.

[13] 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.

[14] 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.

[15] ArmonGo, "Github: Geocoordsfeatsext," accessed: March 31, 2025. [Online]. Available: https://github.com/ArmonGo/GeoCoordsFeatsExt

[16] F. Conrad, M. Mälzer, M. Schwarzenberger, H. Wiemer, and S. Ihlenfeldt, "Benchmarking automl for regression tasks on small tabular data in materials design," *Scientific Reports*, vol. 12, no. 1, p. 19350, 2022.

[17] S. B. Rabbani, I. V. Medri, and M. D. Samad, "Attention versus contrastive learning of tabular data - A data-centric benchmarking," *CoRR*, vol. abs/2401.04266, 2024.

[18] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 6, pp. 7499–7519, 2024.

[19] Z. Qian, R. Davis, and M. van der Schaar, "Synthcity: a benchmark framework for diverse use cases of tabular synthetic data," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[20] N. Erickson, L. Purucker, A. Tschalzev, D. Holzmüller, P. M. Desai, F. Hutter *et al.*, "Tabarena: A living benchmark for machine learning on tabular data," *arXiv preprint arXiv:2506.16791*, 2025.

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

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] M. P. Lucas, R. J. Longman, T. W. Giambelluca, A. G. Frazier, J. Mclean, S. B. Cleveland, Y.-F. Huang, and J. Lee, "Optimizing automated kriging to improve spatial interpolation of monthly rainfall over complex terrain," *Journal of Hydrometeorology*, vol. 23, no. 4, pp. 561–572, 2022.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] B. S. Murphy, "Pykrige: development of a kriging toolkit for python," in *AGU Fall Meeting Abstracts*, vol. 2014, 2014, pp. H51K–0753.

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

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

[38] 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.

[39] J. Pearl, *Causality*. Cambridge University Press, 2009.

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

[41] B. Peters, V. Niculae, and A. F. T. Martins, "Sparse sequence-to-sequence models," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 2019, pp. 1504–1519.

[42] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, pp. 18 932–18 943.

[43] J. Chen, K. Liao, Y. Wan, D. Z. Chen, and J. Wu, "Danets: Deep abstract networks for tabular data classification and regression," in *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 2022, pp. 3930–3938.

[44] L. Anselin, R. Bongiovanni, and J. Lowenberg-DeBoer, "A spatial econometric approach to the economics of site-specific nitrogen management in corn production," *American Journal of Agricultural Economics*, vol. 86, no. 3, pp. 675–687, 2004.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[50] 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.

[51] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge: Cambridge University Press, 2000.

[52] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[53] 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.

[54] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[55] 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.

[56] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Inf. Fusion*, vol. 81, pp. 84–90, 2022.

[57] M. Joseph and H. Raj, "Gandalf: gated adaptive network for deep automated learning of features," *arXiv preprint arXiv:2207.08548*, 2022.

[58] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.
[59] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 4765–4774.