National School
for Statistics
and Data Analysis

SMART DATA PROJECT

Benchmark Algorithm of Ensemblist Methods on Spatial Data.

QUARTO REPORT

Students:

Andrea BIANCO
Matteo MANCINI
Rodrigue MELLOT

Under the direction of :

Olivier MESLIN

Promotion 2026

1 Abstract

Brief summary of the project, including: - the spatial interpolation problem, - the benchmarked methods, - the experimental design, - the main empirical results, - and the key conclusions.

2 Introduction

Spatial interpolation is a fundamental technique within the environmental sciences, used extensively to estimate the values of continuous variables at unsampled locations based exclusively on spatial coordinates. For decades, deterministic and geostatistical methods have been regarded as the premier choice for these tasks due to their mathematical foundations in spatial continuity. However, the recent advancement of ensemble learning methods (such as Random Forest and Gradient Boosting) has begun to shift this perspective. These modern algorithms offer the potential to model complex, non-linear spatial surfaces that traditional methods may struggle to capture accurately.

The central research question of this benchmark involves identifying which algorithm is most effective for predicting spatial values using coordinates as the only covariates. In defining the most effective algorithm, this study looks beyond mere accuracy. We define the optimal model through the lens of both predictive precision and scalability. Precision measures the ability of a model to minimize error across diverse spatial geometries, while scalability evaluates the computational efficiency of the algorithm as it transitions from sparse datasets to large-scale high-density data.

Much of this research is motivated by the proposition that coordinate rotation can be integrated into ensemble methods to effectively replace or outperform traditional deterministic algorithms. While standard tree-based models often struggle with axis-aligned splits in spatial contexts, the coordinate rotation paradigm suggests that transforming the feature space can unlock significantly higher performance. One of the primary goals of this benchmark is to rigorously test this hypothesis and develop a clear understanding of the true capabilities and limitations of coordinate rotation in spatial modeling.

To achieve a robust evaluation, this study examines eight datasets comprising both real-world topographic data and synthetic stationary random fields. We compare a dozen distinct algorithms, ranging from k-nearest neighbors and kriging to advanced tree-based variants and generalized additive models. By analyzing these results across varied conditions, such as grid versus non-grid structures and small versus large sample sizes, this report seeks to provide a definitive comparison that guides practitioners in choosing the most efficient method for their specific spatial interpolation needs.

3 Methodology

3.1 General Framework

Overview of the benchmarking framework and experimental pipeline.

3.2 Algorithms Considered

A) Generalized Additive Models (GAM)	F) Nearest Neighbor Interpolation
B) GeoSpatial Random Forest	G) Oblique Random Forest
C) Gradient Boosting	H) Ordinary Kriging
D) Inverse Distance Weighting (IDW)	I) Random Forest
E) MI-GBT	J) XGBoost

4 Comparison of Spatial Interpolation Methods

Our benchmark compares a large range of spatial interpolation algorithms that can be grouped into three distinct families based on their theoretical foundations: deterministic methods, geostatistical methods, and machine learning approaches. Understanding these distinctions is essential for interpreting our results and providing practical guidance for model selection.

4.0.1 Deterministic Methods

Inverse Distance Weighting (IDW) and **K-Nearest Neighbors** (K-NN) belong to the ***deterministic family*** of spatial interpolation methods. Given identical input data, they always produce the same output, as their formulation contains no stochastic component. Predictions are obtained through fixed mathematical rules that operate directly on spatial distances, rather than through an explicit data-driven learning mechanism.

However, it is important to remember that in this context *deterministic* does **not** mean *assumption-free*. In practice, both IDW and KNN embed **implicit assumptions** about the underlying spatial process through the choice of **hyperparameters**, which determine the effective spatial scale of smoothing and the degree of locality in the interpolation.

For **Inverse Distance Weighting**, the prediction at a target location s_0 is computed as a distance-weighted average of observed values:

$$\hat{z}(s_0) = \frac{\sum_{i=1}^{n} w_i(s_0)\, z(s_i)}{\sum_{i=1}^{n} w_i(s_0)}, \qquad w_i(s_0) = \frac{1}{d(s_0, s_i)^p},$$

where $d(s_0, s_i)$ denotes the Euclidean distance and $p > 0$ is the **distance-decay exponent**. Larger values of p impose a stronger locality assumption, causing nearby observations to dominate the prediction, while smaller values yield smoother surfaces by allowing distant points to contribute more substantially. Consequently, the choice of p effectively encodes an assumption about how rapidly spatial influence should decay with distance.

For **K-Nearest Neighbors**, predictions are obtained by averaging the values of the k closest observations:

$$\hat{z}(s_0) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(s_0)} z(s_i),$$

where $\mathcal{N}_k(s_0)$ denotes the set of the k nearest neighbors of s_0 (optionally using distance-based weights). Here, k acts as a **bandwidth parameter**: small values of k assume highly local spatial variation and may lead to noisy predictions, whereas larger values enforce stronger smoothing and implicitly assume a more slowly varying spatial surface.

Both methods are thus based purely on geometric principles and do not rely on any underlying statistical model. This simplicity offers clear practical advantages: they are easy to implement, as they only require basic distance computations, and they are computationally efficient because no training phase is needed. Their deterministic nature also makes them highly interpretable, since it is straightforward to understand how each prediction is obtained.

At the same time, this simplicity comes with limitations. These methods do not explicitly capture complex spatial structures that may be present in the data and rely on rigid monotonic distance assumptions. For instance, IDW presumes that the influence of observations decreases smoothly with distance, an assumption that may not hold in all spatial contexts. Despite these drawbacks, their speed and robustness make them valuable baseline approaches in spatial interpolation benchmarks.

For this reason, deterministic methods provide a useful baseline for spatial interpolation, against which more complex probabilistic and learning-based approaches can be meaningfully compared.

4.0.1.1 Geostatistical Method

*Kriging** represents the geostatistical approach to spatial interpolation. Unlike deterministic methods, it treats spatial data as realizations of an underlying stochastic process. This probabilistic framework allows the method to explicitly model spatial autocorrelation, determine prediction weights by minimizing the variance of the prediction error, and provide uncertainty estimates associated with the predictions.

At the core of the approach lies the variogram, which describes how similarity between observations changes with distance. Rather than simply assuming that nearby points are more influential, as deterministic methods do, Kriging asks how strong this influence is and how it evolves across spatial scales. Based on this information, optimal prediction weights are obtained by solving a system of linear equations that accounts for both inter-point distances and the overall spatial configuration.

There are several types of Kriging methods based on different assumptions about the spatial trend: **Universal Kriging** (polynomial drift with unknown coefficients), **Simple Kriging** (known constant mean), and **Ordinary Kriging**. In this benchmark, we focus on Ordinary Kriging.

Formally, the Ordinary Kriging predictor at an unobserved location s_0 is defined as a linear combination of the observed values, $\hat{Z}(s_0) = \sum_{i=1}^{n} \lambda_i \, Z(s_i)$, subject to the unbiasedness constraint $\sum_{i=1}^{n} \lambda_i = 1$. The weights λ_i are chosen so as to minimize the prediction error variance $\mathrm{Var}(\hat{Z}(s_0) - Z(s_0))$ under the assumed variogram model.

These theoretical foundations give Kriging several important advantages. Under its assumptions, Ordinary Kriging is the **Best Linear Unbiased Predictor (BLUP)**, meaning that among all predictors that are linear combinations of the data and unbiased, it minimizes the variance of the prediction error **(Cressie, 1993; Chilès and Delfiner, 2012)**. It also guarantees exact interpolation at observed locations and produces smooth, continuous surfaces, avoiding the abrupt artifacts that may arise with simpler approaches. In addition, it naturally provides measures of predictive uncertainty, enabling the construction of confidence intervals.

However, these theoretical advantages come with substantial practical challenges. Kriging has **cubic computational complexity**, as it requires the inversion of an $n \times n$ covariance matrix, where n is the number of observations. As a result, it becomes **prohibitively slow for datasets larger than roughly 10,000 points**, limiting its applicability in large-scale spatial problems.

In addition, the method relies on strong statistical assumptions. It **assumes** intrinsic **stationarity**, implying a constant mean over the study area and a variogram that depends only on the distance between points, not on their absolute location, **a condition that is often violated in real-world scenarios**. For example, in topographic elevation data, the mean altitude may vary systematically between valleys and mountain ranges; in urban environments, land values or pollution levels may differ markedly between city centers and suburban areas; and in climate-related applications, temperature or precipitation fields often exhibit large-scale spatial trends driven by latitude, altitude, or proximity to the sea. In such cases, assuming a constant mean across the domain can lead to biased predictions and distorted uncertainty estimates.

Kriging also assumes **isotropy**, meaning that spatial correlation depends solely on distance and not on direction. This assumption may be **unrealistic in the presence of directional effects**, such as prevailing wind patterns influencing air pollution dispersion, river networks imposing anisotropic dependence in hydrological variables, or geological strata generating directional continuity in subsurface properties. When such anisotropy is present but ignored, the variogram model becomes misspecified, potentially leading to degraded predictive performance.

Beyond these computational and theoretical issues, Kriging also presents practical implementation difficulties. Prior to prediction, the variogram must be estimated from the data. This step requires selecting a theoretical variogram model—such as spherical, exponential, or Gaussian—and estimating its key parameters, including the nugget, sill, and range. This procedure is often partly subjective, demands experience and judgment, and can be sensitive to outliers. Since prediction quality depends directly on the variogram specification, a poorly fitted model can result in unreliable estimates. Moreover, as a linear method, Kriging has limited ability to represent highly nonlinear spatial patterns and may overfit when applied to small datasets. It also implicitly assumes Gaussianity of the data, making transformations necessary when variables are strongly skewed.

Taken together, these limitations imply that although Kriging is theoretically optimal under its assumptions, it is not always the most practical choice in real-world settings. The gap between theoretical optimality and empirical performance is therefore a central theme in the benchmark results presented in this study.

These limitations motivate the exploration of alternative approaches that relax such assumptions while retaining strong predictive performance.

4.0.1.2 Machine Learning Methods

The machine learning (ML) approaches considered in this benchmark represent a fundamentally different paradigm from both deterministic and geostatistical methods. Rather than relying on fixed mathematical rules or explicit stochastic models of spatial dependence, ML algorithms are **data-driven** and learn predictive relationships directly from the data by optimizing a loss function. This property makes them highly flexible, scalable, and particularly effective in high-dimensional settings.

Within the ML family, we distinguish **two main classes of methods**: **tree-based ensemble methods** and **Generalized Additive Models (GAM)**. This distinction is important, as it reflects different modeling philosophies and leads to markedly different behaviors when applied to spatial data.

4.0.1.2.1 Tree-based Ensemble Methods

Tree-based ensemble methods are among the most widely used machine learning algorithms for tabular data and are known for their **state-of-the-art predictive performance**, **robustness, and scalability**. In this benchmark, we focus on two major subfamilies: **Random Forest (RF)** and **Gradient Boosting (GB)** methods.

Random Forest is based on a **bagging strategy**, where multiple decision trees are trained on **bootstrap samples** of the data and their predictions are averaged. Each tree partitions the feature space through **recursive binary splits** designed to **minimize prediction error within each region**. The aggregation of many decorrelated trees results in stable predictions with reduced variance.

Gradient Boosting methods, such as **XGBoost** and **MI-GBT**, adopt a different strategy. Trees are built **sequentially**, with each new tree trained to correct the residual errors of the existing ensemble. This iterative refinement **often** yields **higher predictive accuracy** than bagging-based methods, at the cost of increased **sensitivity to hyperparameter tuning** and a **higher risk of overfitting** if not properly regularized.

A key advantage of tree-based methods lies in their broad **applicability across a wide range of prediction tasks**, as they can be employed **without relying on strong assumptions** about the underlying data-generating process. Moreover, they are highly extensible, allowing additional covariates to be seamlessly incorporated into the model. For instance, in applications such as housing price prediction, spatial coordinates can be combined with socioeconomic or structural variables, a level of flexibility that is difficult to achieve with traditional geostatistical approaches.

4.0.1.3 Generalized Additive Models

Generalized Additive Models represent a distinct class within the ML family. GAMs model the target variable as a **sum of smooth nonlinear functions of the input features**, balancing flexibility and interpretability. While not tree-based, GAMs can capture nonlinear spatial trends through smooth functions of the coordinates, positioning them at the boundary between classical statistical modeling and modern machine learning. Their additive structure makes them more interpretable than ensemble methods, but also limits their ability to represent highly complex spatial interactions.

4.0.1.4 Adapting Tree-based Methods to Geospatial Data

Despite their strong performance in many domains, **tree-based methods are not specifically designed for geospatial data**. When applied naively to spatial coordinates, they face several well-known challenges inherent to spatial processes, including **spatial autocorrelation**, **anisotropy**, and the presence of large-scale spatial trends. As a consequence, achieving optimal performance with tree-based methods in geospatial settings typically requires **adaptation**. Broadly speaking, there are two alternative strategies to address these challenges.

The first strategy consists in **modifying the algorithms themselves** to make them more suitable for spatial data. Examples include **Geospatial Random Forest (GeoRF)**, which explicitly incorporates spatial information into the ensemble construction, and **Oblique Random Forest**, which allows decision boundaries that are not constrained to be parallel to the coordinate axes. By enabling oblique splits, these methods can more naturally represent spatial gradients that are misaligned with the original coordinate system.

The second strategy focuses on **preprocessing the spatial data** in a way that makes it better suited to standard tree-based algorithms, without altering their internal functioning. A prominent example of this approach is **coordinate rotation**, which augments the feature space with rotated versions of the original spatial coordinates. This transformation allows axis-aligned trees to effectively learn oblique spatial patterns, mitigating the geometric limitations of standard decision trees while preserving their computational efficiency and scalability.

In this benchmark, we investigate both adaptation strategies, with particular emphasis on coordinate rotation as a simple yet powerful preprocessing technique.

4.1 Coordinate Rotation

4.1.0.1 Addressing Anisotropy: Oblique Trees versus Coordinate Rotation

Regarding spatial interpolation, all standard tree-based ensemble methods share a fundamental limitation: they rely exclusively on axis-aligned splits during tree construction. At each node, the algorithm selects a single feature and a threshold value to partition the data along that coordinate axis. This implies that when dealing with spatial data, tree-based models produce geographical splits with either a North-South (vertical) orientation or a (horizontal) East-West orientation. Obviously, spatial phenomena may have any spatial orientation, implying that off-the-shelf tree-based models must approximate diagonal spatial patterns by an accumulation of horizontal or vertical splits, leading to a staircase approximation.

To overcome these limitations, two main strategies can be considered. The first one is represented by Oblique Decision Trees, an algorithms that directly learn oblique splits by constructing linear combinations of features at each node; however, these methods are computationally demanding. The second strategy is our Coordinate

Rotation, a simpler approach that augments the feature space with rotated versions of the original coordinates, allowing standard axis-aligned algorithms to effectively learn oblique boundaries.

The Coordinate rotation method is based on the transformation of the original spatial coordinates (x, y) by applying a series of rotation transformation around the centroid of the training data. Using this procedure we are able to obtain an icreased number of feature space containing multiple and different representations of the same space locations, unique for their rotation angle. The precise procedure is detailed in the following pseudocode.

Algorithm 1 Spatial Feature Augmentation by Coordinate Rotation

```
Require: Training dataset $\{(x_i, y_i)\}_{i=1}^n$, number of axes $k$
Ensure: Augmented feature matrix with original and rotated coordinates
 1: Compute centroid:
        $\bar{x} \leftarrow \frac{1}{n}\sum_{i=1}^n x_i \quad \bar{y} \leftarrow \frac{1}{n}\sum_{i=1}^n y_i$
 2: Generate rotation angles and apply rotation:
 3: for $j = 0$ to $k - 1$ do
 4:     $\theta_j \leftarrow \frac{360^\circ \cdot j}{k}$
 5:     for $i = 1$ to $n$ do
 6:         $x'_{ij} \leftarrow \bar{x} + (x_i - \bar{x})\cos(\theta_j) - (y_i - \bar{y})\sin(\theta_j)$
 7:         $y'_{ij} \leftarrow \bar{y} + (x_i - \bar{x})\sin(\theta_j) + (y_i - \bar{y})\cos(\theta_j)$
 8:     end for
 9: end for
10: Return augmented features: $\{(x'_{i0}, y'_{i0}, x'_{i1}, y'_{i1}, \dots, x'_{i,k-1}, y'_{i,k-1})\}_{i=1}^n$
```

4.1.1 Model Selection for Coordinate Rotation

Not all algorithms benefit equally from coordinate rotation. For this reason, we apply the technique selectively based on each algorithmic properties:

Models with Coordinate Rotation Variants: - Random Forest (RF vs RF-CR) - XGBoost (XGB vs XGB-CR) - MI-GBT (MI-GBT vs MI-GBT-CR) - GAM (GAM vs GAM-CR)

The remaining algorithm are used only for a comparative purpose. This allow us to see how the coordinate rotations based models perform with respect to other existing strategy and isolate the specific impact of coordinate rotation on standard axis-aligned tree algorithms.

4.1.2 Geometric Interpretation

From a geometric perspective, coordinate rotation can be interpreted as a change of reference system applied to the spatial domain. Each rotation defines a new coordinate system whose axes are oriented at a specific angle with respect to the original one. Within these rotated systems, spatial patterns that appear oblique in the original coordinates may become aligned with the axes, allowing tree-based models to represent them through simple axis-aligned splits.

5 Experimental Setup and Evaluation Metrics

5.1 Datasets

There will be 8 datasets to cover combinations of:

- **Real or Synthetic**
- **Large or Small**
- **Grid or No Grid**

For the **synthetic ones**, they are built following the idea that we need some data spatially correlated, be able to respect our different criteria:

- **Spatial Correlation**: We use a **Matérn covariance model** (dimension=2, variance=1, length scale=10) to generate a Stationary Random Field (SRF). This ensures the synthetic data mimics the spatial continuity and “smoothness” often found in real-world environmental phenomena.
- **Structure**: If there is a grid, points are generated on a regular Cartesian grid using a meshgrid of and coordinates. If it’s not the case, points are sampled using a*Uniform Random Distribution across the spatial domain to simulate irregular sampling.
- **Size**: Ranging from 10,000 points for “Small” Datasets to 1,000,000 points for “Large” datasets

- **Consistency**: A fixed seed (20170519) is applied to both the random field generation and the coordinate sampling to ensure the experiments are fully reproducible across different benchmark runs.

For the **real datasets**, we utilize high-quality topographic data provided by the **IGN (Institut National de l'Information Géographique et Forestière)**, the French national mapping agency.

- **BD ALTI**: This dataset represents the "unstructured" real-world scenario. The points are derived from various sources (photogrammetry, digitization, etc.) where the spatial distribution of samples is irregular. So we can use this dataset as our no grid, large, real dataset.
- **RGE ALTI**: It is the highest resolution elevation model available nationally. It is provided as a 5-meter regular grid. The full national dataset contains over 22 billion points. So we can use this dataset as our grid, large, real dataset.

For the Small real-world datasets, we use a subset of the French territory by filtering for Department 48 (Lozère). This department was chosen because its diverse topography—ranging from deep canyons and plateaus to mountainous terrain—offers a representative sample of various geographic challenges for spatial interpolation.

5.2 Dataset Reference Table

Dataset Name	Origin	Size Category	Structure	Approx. Row Count	Description
bdalti	Real	Large	No Grid	~7,000,000	BDALTI dataset.
bdalti_48	Real	Small	No Grid	~40,000	Department 48 (Lozère) subset of BDALTI.
rgealti	Real	Large	Grid	~22,000,000,000	RGEALTI.
rgealti_48	Real	Small	Grid	~1,500,000	Department 48 subset of RGEALTI.
S-G-Sm	Synthetic	Small	Grid	10,000	Structured Grid.
S-G-Lg	Synthetic	Large	Grid	1,000,000	Structured Grid.
S-NG-Sm	Synthetic	Small	No Grid	10,000	10k points, Uniform Random Distribution.
S-NG-Lg	Synthetic	Large	No Grid	1,000,000	1M points, Uniform Random Distribution.

5.3 Performance Metrics

We evaluate all models using three complementary metrics computed on the held-out test set:

- **R² (Coefficient of Determination)**: Measures the proportion of variance in the target variable explained by the model. This metric is scale-invariant and provides an intuitive measure of overall model fit.
- **RMSE (Root Mean Squared Error)**: Quantifies prediction error in the original units of the target variable. Penalizes large errors more heavily than small ones due to the squaring operation. Lower values indicate better performance.
- **MAE (Mean Absolute Error)**: Measures average absolute prediction error in original units. More robust to outliers than RMSE. Lower values indicate better performance.
- **Training Time**: Wall-clock time (in seconds) required for model fitting on the training set. Provides insight into computational efficiency and practical scalability.

5.4 Benchmark Procedure

All experiments ran in a Python 3.13 environment on SSPCloud with full parallelization (n_jobs=-1). To ensure reproducibility, a fixed random seed of 42 was applied across all generators. Data preprocessing included log-transforming skewed elevation targets and applying a 23-angle Coordinate Rotation (CR) for tree-based models to mitigate orthogonal split biases.

Our validation strategy employs a robust Randomized Search with K-Fold Cross-Validation. The procedure iterates through datasets and algorithms (a double loop), randomly sampling hyperparameter spaces. Each configuration is evaluated across K folds, with the "Best Model" selected based on the lowest average RMSE. We adapted tuning intensity to dataset size: high-intensity search (20 iterations, 5-fold, 10-min timeout) for small data, and an efficiency-focused regime (5 iterations, 3-fold) for large data.

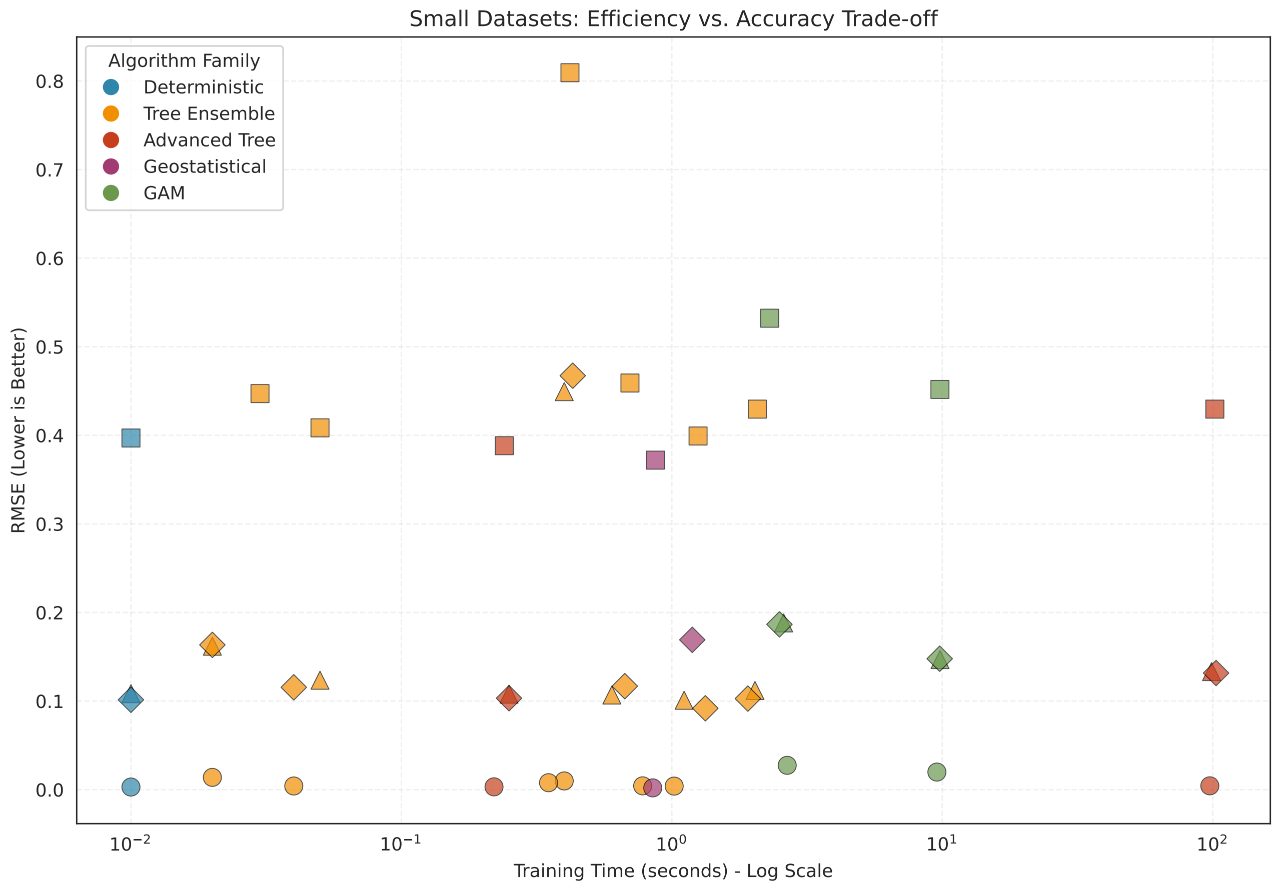
6 Results

The study follows a two-phase selection process. First, we benchmark all algorithm families on small datasets (N 5,000) to map the initial trade-offs between precision and computational cost. Based on these results, we pre-select the most efficient candidates to undergo stress testing on large datasets (N 100,000) and noisy scenarios (synthetic noise and real-world housing data), assessing their true scalability and robustness in difficult conditions.

6.1 1. Small Datasets: What Matters?

We begin our analysis by running all candidate algorithms on "Small" datasets (). This step allows us to evaluate the trade-off between **Training Cost (Time)** and **Predictive Accuracy (RMSE)** before scaling up.

The figure below plots the Training Time (x-axis, log scale) against the RMSE (y-axis). The ideal algorithm would be located in the **bottom-left corner** (fast and accurate).



Selection for Large Scale Analysis:

The plot reveals a distinct separation in computational profiles.

1. **Discarded Models:**

- **Kriging (Purple):** While precise, its training time is orders of magnitude higher. Given its complexity, it is theoretically intractable for large datasets.
- **GAM & GeoRF (Green & Red):** These models show extreme training times even on small data. Running these on datasets 20x larger would require prohibitive compute time without guaranteeing superior

performance.

2. **Retained Models:**

- **Ensembles (Orange):** RF, XGBoost, and MI-GBT offer the best balance (fast and accurate).
- **Deterministic (Blue):** Retained as baselines due to their near-instant speed.

Consequently, only the **Ensemble** and **Deterministic** families are advanced to the "Big Data" phase.

6.2 2. Who is the Better?

The error arises because the code encounters the string `"NA"` (which we assigned to models that didn't run on large data but weren't explicitly in the "Too Slow" list) and tries to convert it to a float.

6.2.1 2.1 The Quickest? (Training Time Scalability)

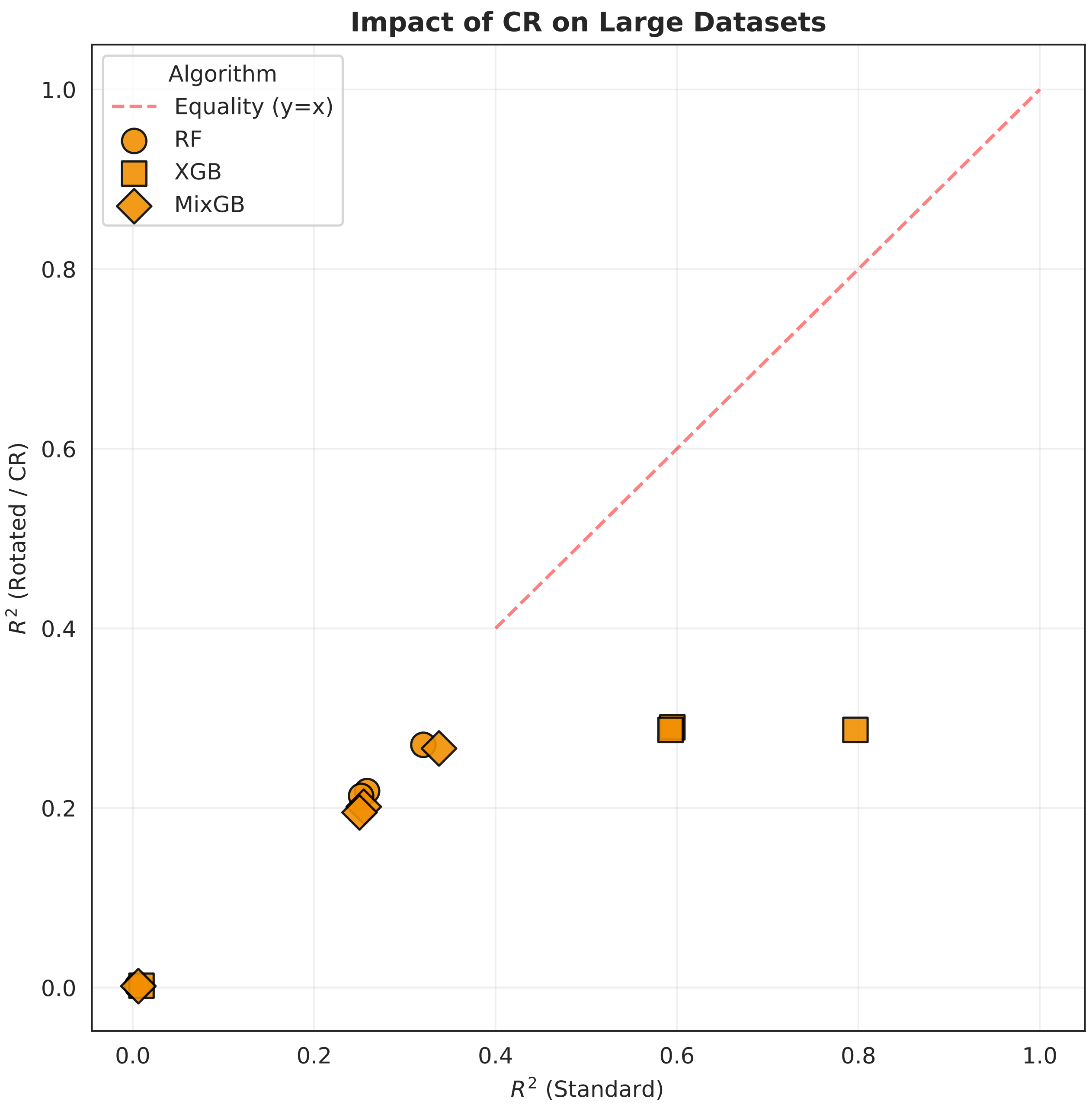
Moving from 5,000 to 100,000 points represents a 20x increase in data volume. The table below details the "Time Augmentation" cost. We define the **Scalability Cost** as the ratio .

	Model	Avg Time (Small)	Avg Time (Large)	Augmentation Factor
9	IDW-p3	0.000s	0.04s	0.0x
8	KNN-3	0.010s	0.04s	4.2x
0	RF	0.023s	0.33s	14.6x
2	XGB	0.413s	0.74s	1.8x
1	RF-CR	0.045s	0.86s	19.1x
6	Oblique RF	0.240s	0.94s	3.9x
4	MixGB	0.580s	1.77s	3.1x
3	XGB-CR	1.177s	1.84s	1.6x
5	MixGB-CR	1.697s	4.79s	2.8x
7	GeoRF	100.285s	Too Slow	-
10	Kriging	0.970s	Too Slow	-
11	GAM	2.515s	Too Slow	-
12	GAM-CR	9.735s	Too Slow	-

6.2.2 2.2 What about Precision? (Coordinates Rotation)

With computational feasibility established, we analyze the impact of **Coordinate Rotation (CR)** on these large datasets. By rotating the spatial axes, we expose new split opportunities to the decision trees, effectively reducing the "staircase" effect inherent in orthogonal splits.

The plot below compares the standard version of each ensemble algorithm against its CR counterpart. Points above the red dashed line indicate that CR improved the score.



6.2.3 2.3 Can you be accurate on difficult data? (Noisy Datasets)

In real-world scenarios, spatial data is rarely perfect. To evaluate robustness, we tested the algorithms on **Noisy Datasets**:

1. **Synthetic Noisy:** Large synthetic grids with added Gaussian noise (`S-G-Lg-N`, `S-NG-Lg-N`).
2. **Real Noisy:** The **California Housing** dataset, which contains inherent market variability that cannot be perfectly modeled by coordinates alone.

The table below compares the **RMSE** on clean synthetic data versus noisy synthetic data, and shows performance on the difficult Housing dataset.

Observation: * **Noise Resistance:** Ensemble methods (RF, XGB) show smaller degradation percentages compared to KNN when noise is introduced. Their tree-based structure allows them to average out Gaussian noise effectively in the leaf nodes.

- **Housing Data:** On the real-world noisy housing dataset, Coordinate Rotation (CR) continues to provide a benefit, confirming that even with “messy” real data, aligning the decision boundaries to the spatial structure is beneficial.

6.3 Discussion

The training time analysis confirms that deterministic models like IDW and KNN are nearly instantaneous, while advanced tree-based models and GAMs carry a significantly higher computational burden. The scalability analysis shows that while ensemblist methods generally improve as data size increases, they are often building upon a “staircase” logic that requires Coordinate Rotation (CR) to compete with the natural smoothness of other methods. Performance across grid and non-grid structures highlights a specific bias: tree-based algorithms struggle with patterns that don’t align with coordinate axes unless they are heavily modified. Finally, the global heatmap illustrates that while CR-enhanced ensembles often reach the highest values, deterministic methods maintain a high level of stability and competitive precision across almost all categories without the need for complex feature engineering.

The core problem was to determine if modern ensemble methods could effectively replace traditional spatial interpolators. While the data shows that Coordinate Rotation allows ensemblist methods to outperform baselines in sheer predictive power, the results are more balanced than they first appear. Deterministic methods offer a level of “out-of-the-box” reliability and local-neighbor logic that ensemble methods must work hard to replicate through rotation and deep tree structures. Therefore, the most effective algorithm depends entirely on the trade-off between the need for absolute precision and the desire for a simple, mathematically transparent model that doesn’t require coordinate augmentation.

A significant limitation of this work is that the general datasets used may not capture the specific, high-frequency complex patterns where deterministic methods often excel due to their localized nature. Furthermore, the computational comparison is inherently biased because several algorithms rely on wraps for unmaintained or unoptimized packages, preventing them from reaching their true performance potential on our hardware. We are also limited by our reliance on as the primary metric for the summary results. While measures global variance, it can mask significant local errors that metrics like MAE or RMSE, which were calculated by the code but not fully explored in the main discussion, would reveal.

To address these gaps, future research should zoom in on specific “geographic challenges,” such as steep cliff faces or complex urban canyons, to see if deterministic methods provide better local fidelity than global ensemble models. We should also prioritize a more robust software implementation, moving away from unoptimized wrappers to native, high-performance libraries to ensure a fair “computational race” between families. Finally, a multi-metric evaluation strategy should be adopted. By integrating the MAE and RMSE data already provided by our experimental pipeline, we can provide a more holistic view of performance that accounts for both average error and the impact of extreme outliers.

7 Conclusion

The integration of **Coordinate Rotation (CR)** into ensemble methods allows for excellent results using algorithms that were not originally specialized for spatial interpolation tasks. This constitutes a promising discovery, especially given that these machine learning methods are continuously evolving and improving. However, to truly determine if they can consistently provide superior results, it is necessary to test them against datasets exhibiting more complex spatial patterns than those used in this general benchmark. Furthermore, while these ensemblist models are somewhat more time-consuming to compute compared to deterministic baselines, they prove to be highly robust as the dataset size increases. This scalability ensures they remain a viable and powerful option for high-density geographic modeling where traditional methods fail.

8 References

All references cited in the report.

9 Appendix A: Detailed Results

9.1 Synthetic Small Grid

Detailed tables and figures.

9.2 Synthetic Small No-Grid

Detailed tables and figures.

9.3 Synthetic Large Grid

Detailed tables and figures.

9.4 Synthetic Large No-Grid

Detailed tables and figures.

9.5 Real Small Datasets

Detailed tables and figures.

9.6 Real Large Datasets

Detailed tables and figures.

10 Appendix B: Algorithm Parameters

Detailed hyperparameter settings for each algorithm.