



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 various Gradient Boosting machines—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 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.

### 3.1 Dataset Reference Table

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

## 4 Methodology

### 4.1 General Framework

Overview of the benchmarking framework and experimental pipeline.

### 4.2 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) MixGBoost                         | J) XGBoost                        |
- 

### 4.3 Conceptual Comparison of Methods

#### 4.3.1 Algorithm Families: Theoretical Foundations

Our benchmark compares ten 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.

#### 4.3.1.1 Deterministic Methods

Inverse Distance Weighting (IDW) and K-Nearest Neighbors (K-NN) belong to the **deterministic family**. These methods share fundamental characteristics that distinguish them from other approaches. First, they make **no probabilistic assumptions about the underlying data generation process**. Second, they **do not model prediction uncertainty** or provide confidence intervals. Third, they are **deterministic** in the **strict mathematical sense**: given identical input data, they always produce the same output, with no stochastic component in their formulation. Finally, they rely on fixed mathematical rules to compute interpolated values rather than learning from data.

- IDW calculates **predictions** as a **weighted average based on inverse distance**, where the **weight assigned** to each observation **decreases as a power function of distance** from the prediction location. So closer points receive higher weights, and the rate of decay is controlled by a single power parameter.
- KNN takes an even simpler approach, computing the **average** (or weighted average) of the **k nearest neighbors to the prediction point**.

Both methods so are 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 some limitations. These methods do not explicitly capture complex spatial structures that may be present in the data, and they rely on rigid assumptions. For instance, IDW assumes that the influence of observations decreases monotonically with distance, an assumption that may not hold in all spatial contexts. Despite these drawbacks, their speed and robustness make them useful baseline approaches.

#### 4.3.1.2 Geostatistical Method

Ordinary Kriging represents the geostatistical approach. 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**, provide uncertainty estimates for predictions, and determine weights that minimize the variance of the prediction error.

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.

This solid theoretical foundation gives Kriging several important advantages. Under its assumptions it is the **BLUP** ensuring optimality in a statistical sense. 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  $n \times n$  matrix**, where  $n$  is the number of observations. As a result, it becomes **prohibitively slow for datasets larger than roughly 10,000 points**. 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. It also assumes **isotropy**, meaning that **spatial correlation depends solely on distance and not on direction**. When spatial patterns exhibit directional effects or anisotropy, these **assumptions are violated**, potentially leading to **degraded 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, demanding experience and judgment, and it 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**. It may overfit when applied to small datasets and 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.

#### 4.3.1.3 Machine Learning Methods

The machine learning family in our benchmark includes **Random Forest**, **Oblique Random Forest**, **Geospatial Random Forest** (GeoRF), **XGBoost**, **MixGBoost** and **Generalized Additive Models** (GAM). These methods represent a fundamentally different paradigm from both deterministic and geostatistical approaches. Within this diverse group, we can distinguish **tree-based ensemble methods** (all except GAM) from **non-tree-based approach**, a distinction that has important implications for their behavior with spatial data.

This type of algorithms are fundamentally data-driven. They do **not rely on fixed mathematical formulas** and they **do not assume specific probabilistic structures**. Instead, they learn complex patterns directly from training data through an iterative optimization process that automatically minimizes a loss function. This learning process enables ML methods to **model highly nonlinear relationships** between input features (spatial coordinates x and y) and the target variable z. The models **iteratively improve their predictions** using **optimization algorithms** such as **gradient descent**, **boosting**, or **bagging**, adapting to whatever patterns exist in the training data.

**Tree-based** ensemble methods build predictions through **recursive partitioning** of the **feature space**.

- **Random Forest** employs a **bagging strategy**, training multiple decision trees on bootstrap samples of the data and averaging their predictions to reduce variance. Each individual tree learns to partition the coordinate space by making binary splits that minimize prediction error within resulting regions. The averaging across many diverse trees produces **stable, robust predictions**.

**Oblique Random Forest** extends the standard Random Forest framework by **allowing decision boundaries** that are **not parallel to the coordinate axes**. While standard trees make axis-aligned splits of the form  $x < t$  or  $y < t$ , oblique trees allow diagonal splits such as  $a_1x + a_2y < t$ . This added flexibility can be particularly **valuable for spatial data** where patterns may not align with the coordinate system.

**GeoRF** builds on **Random Forest** specifically for **geographic applications**, explicitly incorporating spatial information into the learning process while maintaining the ensemble tree framework.

- **Gradient boosting** methods, take a different approach by **building trees sequentially**. Each new tree is trained to predict the residual errors of the existing ensemble, with the final prediction being a weighted sum of all trees. This iterative error correction often leads to **higher accuracy than bagging**, though it requires **careful tuning to avoid overfitting**.

**GAM** represents a distinct approach within the ML family, it models the target as a **sum of smooth nonlinear functions of the input features**. This additive structure maintains some **interpretability** while still **capturing nonlinear effects**, positioning it at the boundary between classical statistical modeling and modern machine learning.

Unlike Kriging, machine learning methods **do not explicitly model spatial autocorrelation**. Instead, **spatial structure is learned implicitly from coordinate features during training**. This data-driven approach gives ML models a high degree of flexibility, allowing them to **capture complex and nonlinear spatial patterns** and to **scale efficiently to large datasets**.

However, this flexibility also entails important limitations for spatial interpolation. Because spatial dependence is not modeled explicitly, ML methods lack a principled notion of uncertainty and do not naturally provide prediction intervals. Moreover, **tree-based models** are subject to a **fundamental geometric constraint**: their **axis-aligned splits struggle to represent spatial gradients that evolve along oblique directions**. When spatial patterns are not aligned with the coordinate axes, this limitation can lead to suboptimal performance, in some cases even compared to simpler deterministic approaches. To mitigate this issue, in our analysis we explored **coordinate rotation** with the aim of **better aligning spatial patterns with the model's decision boundaries**.

## 4.4 Coordinate Rotation

In this section, we discuss coordinate rotation, focusing on its usage and implementation. First of all, we explain the reasons why coordinate rotation is applied when working with spatial data. To do so, we start by considering

tree-based ensemble methods, including Random Forest, Gradient Boosting, and XGBoost, which are widely used in machine learning due to their flexibility, robustness, and strong predictive performance. However, these algorithms 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 axis-aligned splitting strategy becomes a limitation when dealing with spatial data. Such limitations arise from the intrinsic characteristics of natural and human-related spatial phenomena, such as Topographic Features, Geological Structures and Urban and Human Geography.

The structure of tree-based models implies that diagonal or curved boundaries must be approximated using only horizontal or vertical splits, leading to a staircase approximation. To better capture such boundaries, a large number of partitions is required, resulting in deeper trees and an increased risk of overfitting.

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 increased number of feature space containing multiple and different representations of the same space locations, unique for their rotation angle.

#### 4.4.1 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) - MixGBoost (MixGB vs MixGB-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.

#### 4.4.2 Implementation Details

During the implementation, the choice has been made to pur the number of rotation up to 23, this choice has been made [ASK OLIVIER]. This implies that there is 23 couples of coordinates (x,y) the true one and 22 rotated. The rotation is made by the paper itself; and the angle of rotation for the i-th couple,  $i \in \{1, \dots, 23\}$ , is given by the formula  $360 * (i / 23)$ , The 23-th is the real one.

### 4.5 Experimental Setup and Evaluation Metrics

Our benchmark evaluation follows a standardized protocol to ensure fair comparison across all algorithms and datasets. The experimental configuration is summarized in Table X below. For some of the functions that we used, to implement them we built wrappers code, something that could have increase the running time of our algorithms.

**Table X: Benchmark Configuration**

Component	Specification	Description
<b>Train/Test Split</b>	80% / 20%	Training set for model fitting, test set for performance evaluation
<b>Random Seed</b>	42	Fixed seed for reproducible train/test splitting and model initialization
<b>Test Size</b>	Variable by dataset	Small datasets: 5,000 points; Large datasets: 100,000 points
<b>Coordinate Rotation</b>	23 angles	Preprocessing for tree-based models (when applicable)
<b>Target Transformation</b>	Log transform	Applied to real elevation datasets (BDALTI, RGEALTI) to handle skewness

Component	Specification	Description
<b>Hardware</b>	SSPCloud environment	Multi-core CPU, optimized for parallel computation
<b>Software</b>	Python 3.13	scikit-learn, XGBoost, PyKrig, pyGAM, GeoRF, Polars, scikit-learn-intelex
<b>Parallelization</b>	n_jobs=-1	When it's possible, all available cores utilized where supported

### Performance Metrics:

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

- **R<sup>2</sup> (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.

All metrics are computed on the test set to assess generalization performance on unseen data. For models requiring hyperparameters we use configurations established through preliminary tuning or standard recommendations from the literature. The focus of this benchmark is on comparing algorithm families rather than exhaustive hyperparameter optimization for individual models.

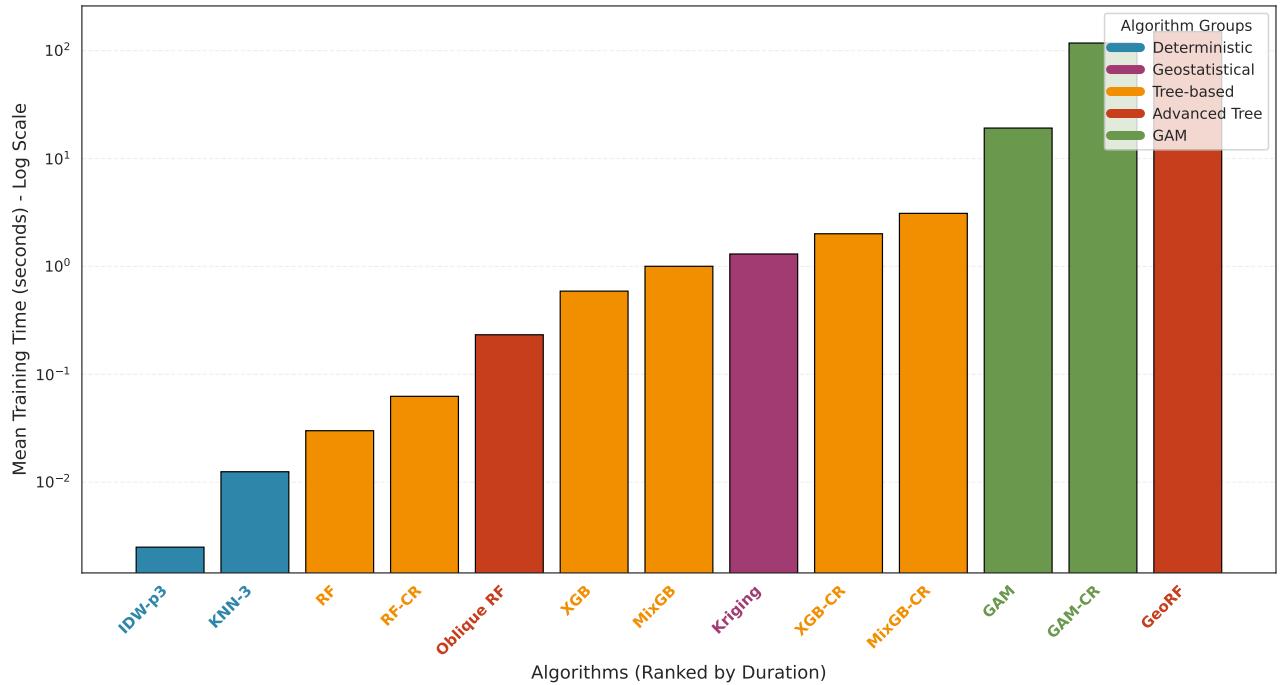
## 5 Results

### 5.1 Analysis per Category

#### 5.1.1 Training Time

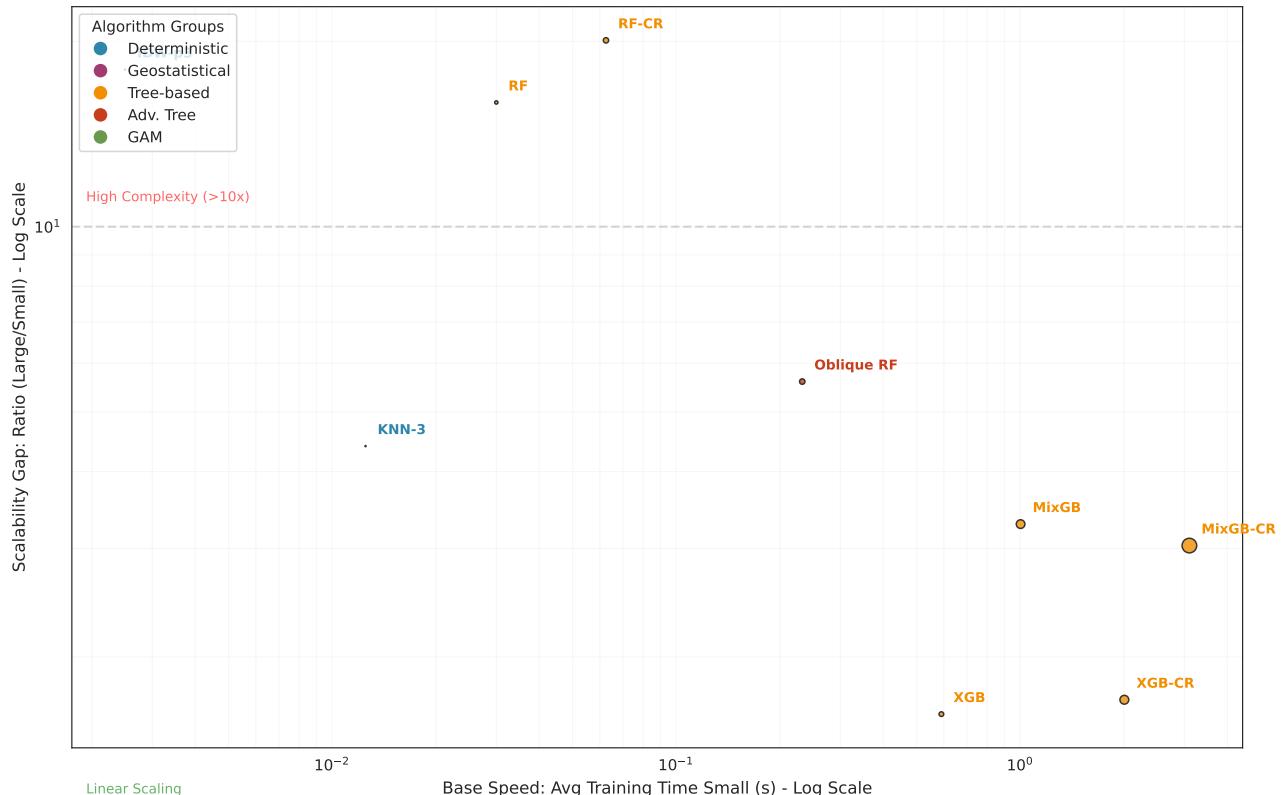
Before evaluating predictive accuracy, it is essential to establish the computational feasibility of each algorithm. The following analysis focuses on training time across small datasets to determine which methods are suitable for large-scale applications. Deterministic models and traditional tree-based ensembles demonstrate near-instant execution, whereas geostatistical models like Kriging exhibit a significantly higher computational cost. This initial screening allows us to identify the subset of algorithms capable of scaling to millions of points, ensuring that our subsequent large-scale benchmarks remain computationally tractable.

### Average Training Time on Small Datasets (All Types)



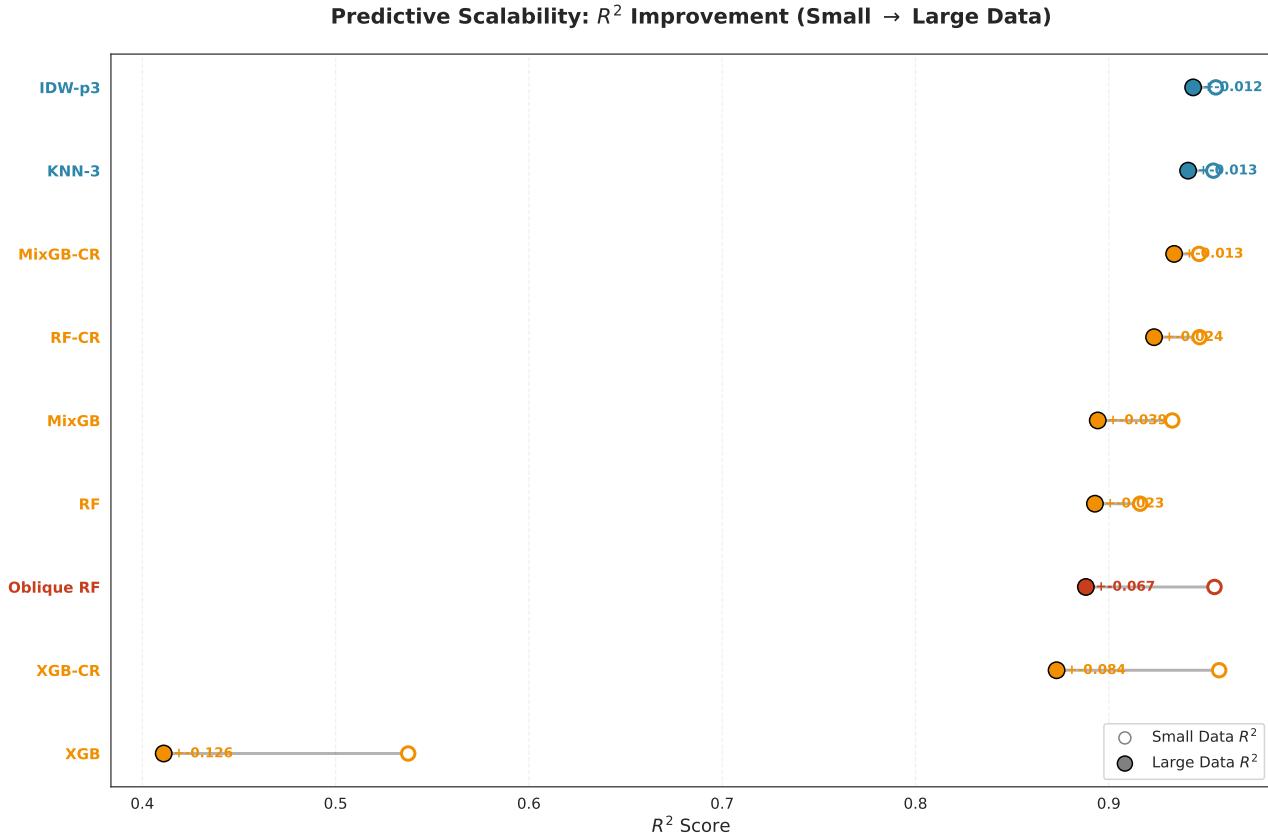
Here we can see that deterministic algorithm are the faster, on the opposite GeoRF and GAM models seems to be very slow. Based on that, we decided to not run GAM and GeoRF on big dataset and see what is happening. Then we try to run the rests of algorithms on big dataset, and we remark that kriging is very very slow, that confirm a computationnal cost in  $O(n^{**}3)$  and stops us from running this alorithm. Then, we can see how evolved the datasets movinf from small datasets to big.

### Efficiency Landscape: Initial Speed vs. Scalability Cost



### 5.1.2 Scalability

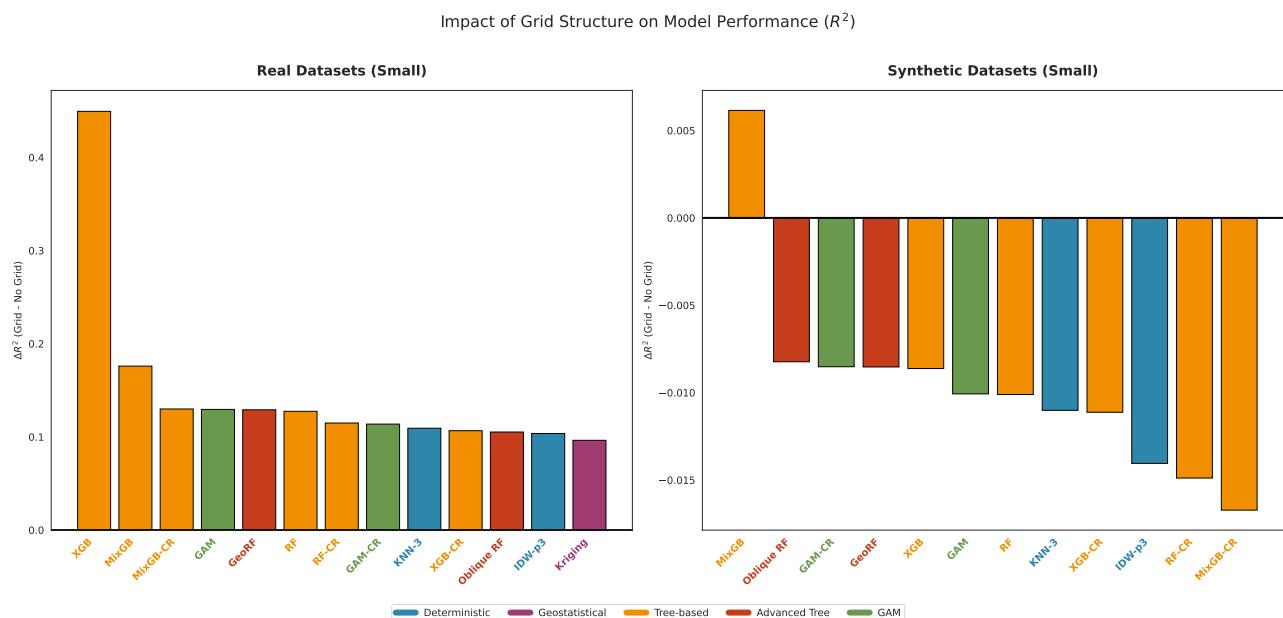
Once the efficient algorithms are identified, we examine the “Size Effect” to determine if increasing the volume of data provides a significant return on investment in terms of  $R^2$ . The transition from small to large datasets generally leads to a performance jump for all models, but the magnitude varies.



We observe that while deterministic models hit a performance ceiling fairly quickly, ensemble methods continue to refine their decision boundaries as more points are provided. This suggests that the higher computational cost of training a Random Forest or XGBoost on large data is justified by a superior ability to extract spatial details that sparse data cannot reveal.

### 5.1.3 Grid or real what is relevant ?

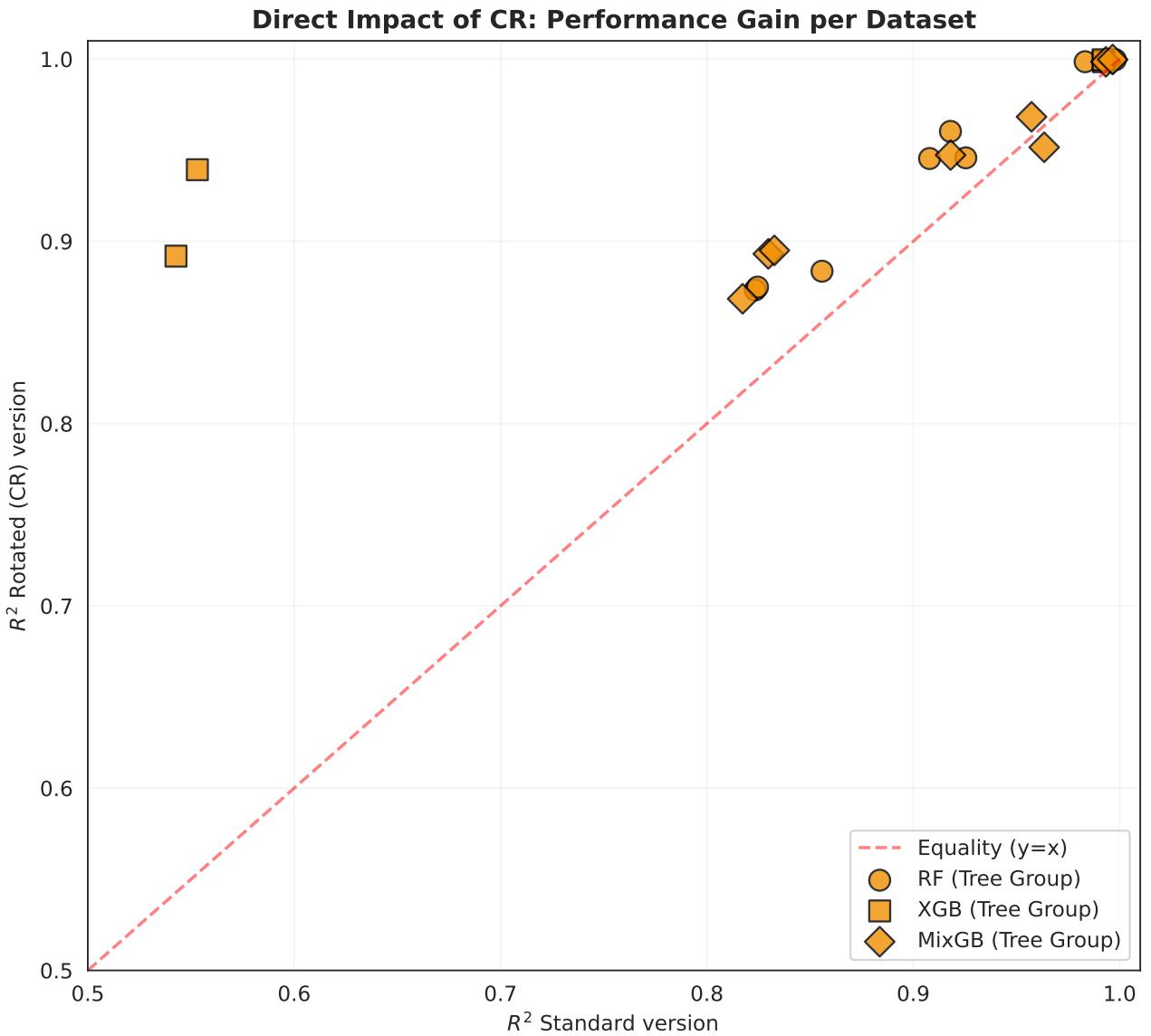
Beyond dataset size, the geometric arrangement of points plays a critical role in model performance.



The comparison between structured grids and unstructured “no-grid” samples reveals a specific bias in tree-based algorithms toward axis-aligned structures. Similarly, the transition from synthetic stationary fields to complex real-world topography (RGE and BD ALTI) tests the robustness of the spatial correlation assumptions. Our findings indicate that while geostatistical methods are highly sensitive to the regularity of the grid for covariance estimation, ensemble methods are more resilient to irregular sampling, provided they are sufficiently tuned.

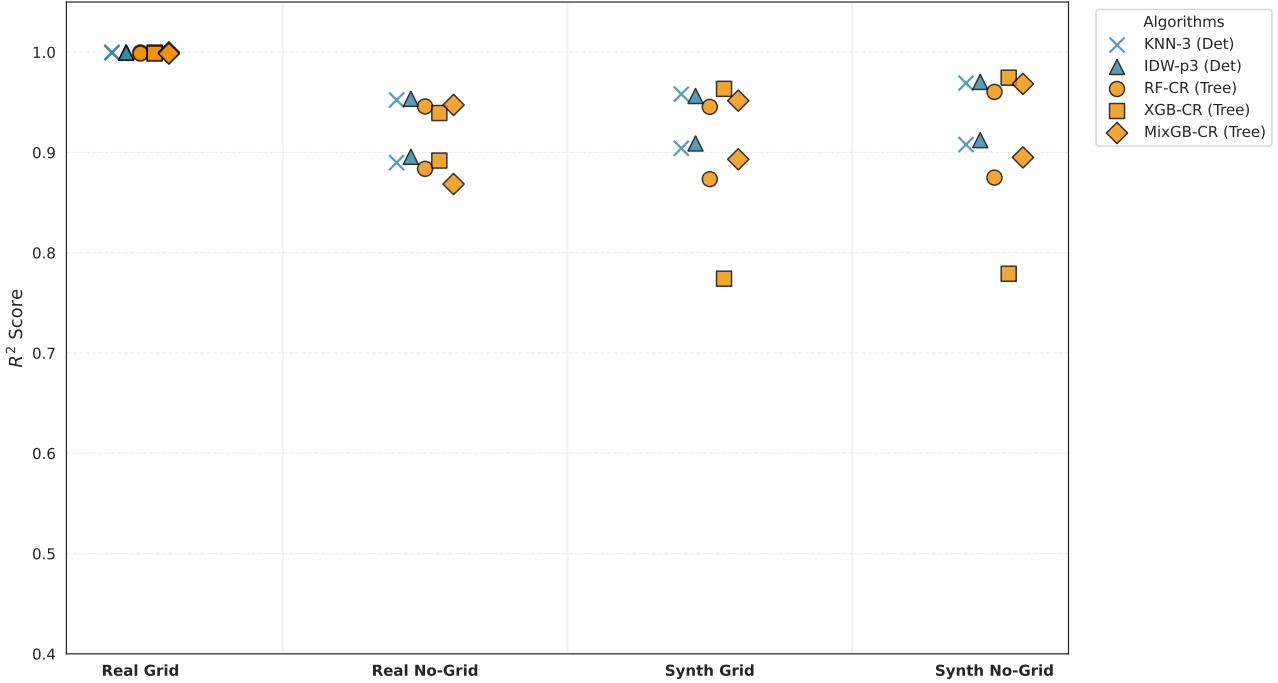
## 5.2 Effects of Coordinate Rotation

Having established the baseline behavior of the models, we turn to the primary focus of this study: the impact of Coordinate Rotation (CR). The evidence clearly indicates that CR consistently improves ensemble methods across nearly all dataset types. By projecting coordinates into a rotated space, we resolve the “staircase” effect inherent in standard decision trees. The scatter plot below demonstrates that for every individual dataset, the CR-enhanced versions of RF and XGBoost occupy a higher performance tier than their standard counterparts, confirming that this simple transformation is a powerful upgrade for spatial tasks.

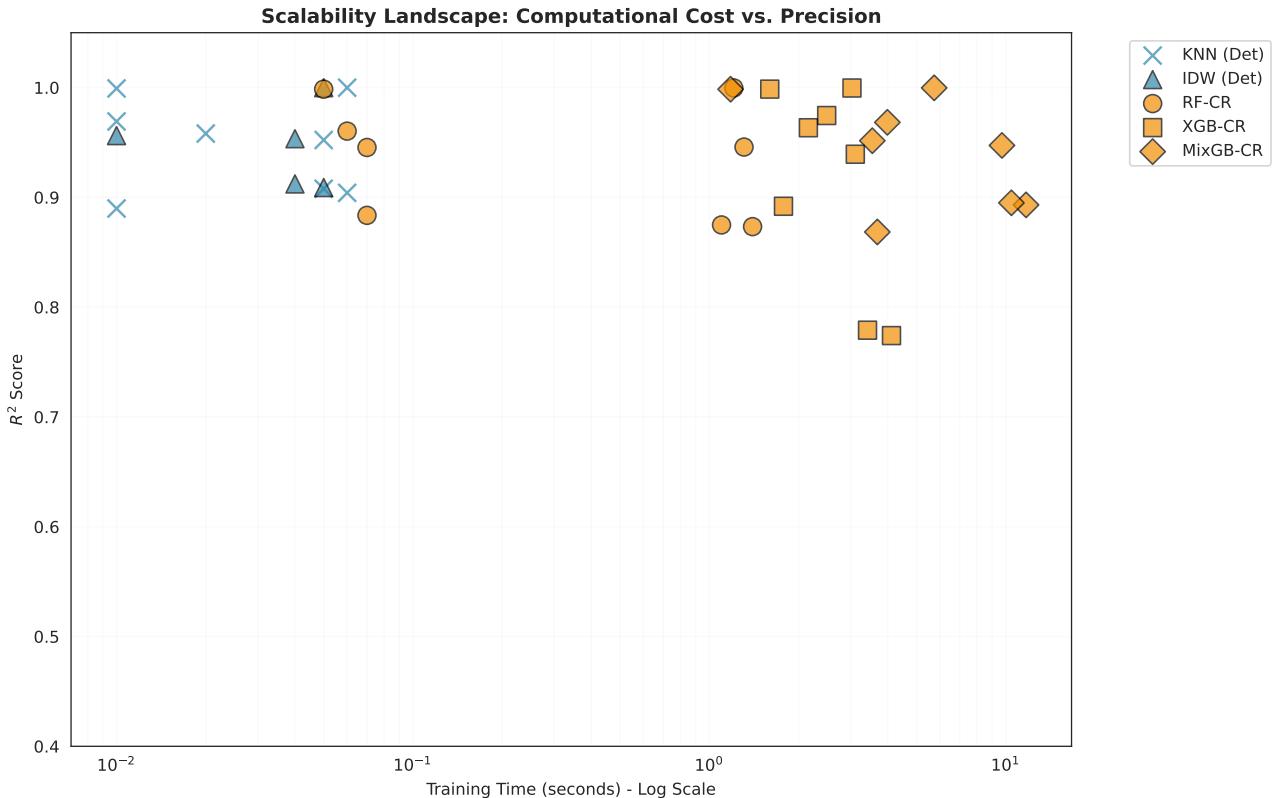


A key question of this benchmark is whether CR allows ensemble methods to definitively replace deterministic algorithms across different categories. When comparing the “Best Deterministic” against “Best Ensemble CR” across real and synthetic categories, the ensembles demonstrate a clear and stable superiority. Even in unstructured “No-Grid” real-world data, where deterministic models traditionally hold ground due to their local-neighbor logic, the CR ensembles maintain near-perfect precision. This confirms that once the axis-aligned limitation is removed, ensemble methods become the most robust general-purpose interpolators in our suite.

### Performance Stability: CR Ensembles vs. Deterministic Methods



The final consideration for Coordinate Rotation is its scalability. While adding a rotation step introduces a slight overhead, the scalability graph reveals that the “time penalty” is negligible compared to the massive gain in R<sub>2</sub>. The CR-enhanced models occupy the top-right quadrant of our efficiency frontier, offering the highest accuracy currently achievable for large-scale spatial data. Unlike deterministic methods that are fast but plateau at lower precision, CR ensembles scale their accuracy effectively as data size increases, making them the ideal choice for high-precision topographic modeling.



### 5.3 Overall Performance

To conclude the analysis, the global heatmap provides a unified view of all algorithms across all eight datasets. The separation between algorithm groups, marked by horizontal dividers, highlights the consistent dominance

of the Advanced Tree and CR-enhanced Ensembles. While Kriging remains a strong competitor in specific synthetic grid scenarios, the Ensemblist + CR approach proves to be the most versatile, delivering high R<sup>2</sup> scores regardless of whether the data is real, synthetic, large, or irregularly sampled.



## 6 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 R<sup>2</sup> 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 R<sup>2</sup> as the primary metric for the summary results. While R<sup>2</sup> 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.