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

This report provides a comprehensive benchmark…

2 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.

3 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.

4 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.

4.1 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 Methodology

5.1 General Framework

Overview of the benchmarking framework and experimental pipeline.

5.2 Algorithms Considered

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

5.3 Conceptual Comparison of Methods

5.3.1 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.

5.3.1.1 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. The formula is straightforward: 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 follow purely geometric rules without any underlying statistical model.

This simplicity brings clear practical advantages. Deterministic methods are easy to implement, requiring only basic distance calculations. They are computationally efficient since no training phase is needed—predictions can be made immediately from the observed data. The transparency of their mechanisms makes them highly interpretable: users can easily understand why a particular prediction was made. However, these strengths come with limitations. Deterministic methods do not explicitly model the complex spatial structure that may exist in the data. They also make rigid assumptions—for instance, IDW assumes that influence always decreases monotonically with distance, which may not hold in all spatial contexts. Despite these constraints, their robustness and speed make them valuable baseline methods.

5.3.1.2 Geostatistical Methods

Ordinary Kriging represents the geostatistical approach, which is rooted in the theory of regionalized variables. Unlike deterministic methods, Kriging treats spatial data as realizations of an underlying stochastic spatial process. This probabilistic framework enables the method to explicitly model spatial autocorrelation, provide uncertainty estimates for predictions, and optimize weights to minimize prediction error variance.

The heart of the Kriging approach is the variogram, a function that quantifies how the similarity between observations changes with distance. Rather than simply assuming that nearby points matter more (as deterministic methods do), Kriging asks a more nuanced question: exactly how much do nearby points matter, and how does this relationship change across different spatial scales? The variogram provides the answer by modeling the spatial correlation structure directly from the data. Through this analysis, Kriging determines optimal prediction weights by solving a system of linear equations that accounts for both the distances to observation points and the overall spatial configuration of the data.

This rigorous theoretical foundation gives Kriging several important strengths. It is proven to be the Best Linear Unbiased Predictor (BLUP) under its assumptions, meaning it provides optimal predictions in a well-defined statistical sense. The method explicitly models spatial structure rather than relying on simple distance-based rules. It guarantees exact interpolation at observed locations—the predicted value at any observation point equals the observed value. The resulting surfaces are smooth and continuous, avoiding the discontinuities that can occur with simpler methods. Perhaps most importantly, Kriging provides uncertainty quantification: it estimates not just the predicted value at each location, but also the prediction variance, enabling the construction of confidence intervals.

However, these theoretical advantages come with significant practical challenges. Kriging has cubic computational complexity, requiring the inversion of an n×n matrix where n is the number of observations. This makes it prohibitively slow for datasets exceeding roughly 10,000 points. The method also rests on strong statistical assumptions. Ordinary Kriging assumes intrinsic stationarity, meaning that the mean is constant (though unknown) across the study area and that the variogram depends only on the separation distance between points, not on their absolute locations. It further assumes isotropy, where spatial correlation depends only on distance, not direction. When spatial patterns show directional trends or anisotropy, these assumptions are violated, potentially degrading performance.

Beyond computational and theoretical constraints, Kriging faces practical implementation challenges. Before making predictions, one must first estimate the variogram from the data. This process involves choosing a theoretical variogram model (spherical, exponential, Gaussian, or others) and estimating its parameters (nugget, sill, and range). This estimation step is often somewhat subjective, requiring judgment and expertise, and the results can be sensitive to outliers in the data. An incorrectly specified variogram directly translates to poor predictions. Additionally, as a linear method, Kriging struggles to capture highly nonlinear spatial patterns. It can produce overfitted results when working with small datasets, and it implicitly assumes that the data follow a Gaussian distribution, requiring transformations when dealing with skewed variables.

These practical limitations mean that while Kriging is theoretically superior in many respects, it is not always the most practical choice for real-world applications. The gap between theoretical optimality and practical performance is an important theme in our benchmark results.

5.3.1.3 Machine Learning Methods

The machine learning family in our benchmark includes Random Forest, Oblique Random Forest, Geospatial Random Forest (GeoRF), XGBoost, HistGradientBoosting, 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 approaches, a distinction that has important implications for their behavior with spatial data.

Machine learning algorithms are fundamentally data-driven. They do not rely on fixed mathematical formulas or 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, without requiring users to specify the functional form in advance. 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. Gradient boosting methods, including XGBoost, HistGradientBoosting, and MixGBoost, 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.

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 splits like "if x $<$ threshold" or "if y $<$ threshold", oblique trees can make diagonal splits such as "if a $\cdot$ x + a $\cdot$ y $<$ threshold". 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.

GAM represents a distinct approach within the ML family. Rather than using trees, 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 GAM at the boundary between classical statistical modeling and modern machine learning.

Critically, none of these ML methods explicitly model spatial autocorrelation in the way that Kriging does. They do not estimate variograms or make assumptions about stationarity. Instead, they learn spatial patterns implicitly through the coordinate features provided during training, discovering whatever structure exists in the data through the optimization process.

This data-driven flexibility brings substantial advantages. ML methods can adapt to virtually any pattern in the data without requiring users to specify functional forms or make strong distributional assumptions. They scale well to large datasets—Random Forest and XGBoost can handle hundreds of thousands of observations efficiently. They automatically capture complex interactions between features, including nonlinear spatial patterns that would be difficult to specify manually. These methods represent the state-of-the-art in many prediction tasks across diverse domains.

However, this flexibility also introduces challenges. ML methods require a distinct training phase with careful attention to train-test splitting and cross-validation to avoid overfitting. They involve numerous hyperparameters (tree depth, number of estimators, learning rates, regularization strengths) that must be tuned, often requiring extensive experimentation. The resulting models are often difficult to interpret—understanding why a Random Forest with 100 trees makes a particular prediction is far less straightforward than understanding why IDW assigns certain weights. Unlike Kriging, ML methods do not naturally provide uncertainty estimates, though techniques like quantile regression or bootstrapping can be applied at additional computational cost.

Most importantly for spatial interpolation, tree-based ML methods face a fundamental geometric challenge. Decision trees make splits parallel to the coordinate axes, which works well when patterns align with these axes but struggles when spatial phenomena exhibit diagonal gradients. Elevation may increase from southwest to northeast,

geological features may be oriented obliquely to map coordinates, or meteorological patterns may follow prevailing wind directions that do not align with north-south or east-west axes. Without additional preprocessing to address this limitation, tree-based methods can substantially underperform even simple deterministic approaches on spatial data.

5.3.2 Synthesis: Choosing Among Families

The three algorithm families represent distinct trade-offs between theoretical rigor, computational efficiency, and practical flexibility. Deterministic methods offer unmatched simplicity and speed at the cost of limited sophistication. Geostatistical methods provide optimal predictions and uncertainty quantification under specific assumptions, but demand computational resources and statistical expertise. Machine learning methods deliver maximum flexibility and can capture complex patterns, but require careful training, tuning, and—for spatial applications—thoughtful preprocessing.

Our benchmark reveals that these theoretical distinctions translate into tangible performance differences that depend critically on the characteristics of the data and the appropriateness of preprocessing. The relationship between algorithmic complexity and predictive performance is not monotonic—more sophisticated methods do not automatically outperform simpler ones. Understanding when each family excels requires examining not just their theoretical properties, but their behavior on real spatial interpolation tasks.

5.4 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 applyng 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 represantions of the same space locations, unique for their rotation angle

5.4.1 Algorithmic Procedure

We applied the coordinate rotation following the steps described below.

Step 1: Compute the Spatial Centroid

For a training dataset with n observations and coordinates (x_i, y_i) for $i = 1, \dots, n$, we first compute the centroid:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

The choice of rotating around the centroid rather than around the origin is made to ensure numerical stability and to preserve the spatial distribution of the data.

Step 2: Define Rotation Angles

We generate a set of uniformly distributed rotation angles. For k total axes (including the original coordinates), we use $k - 1$ rotation angles:

$$\theta_j = \frac{360° \cdot j}{k}, \quad j = 1, 2, \ldots, k-1$$

In our implementation, we use $k = 23$ axes, yielding 22 rotation angles uniformly distributed across the full circle. This choice enables to have a sufficient angular resolution to capture patterns while keeping the computational cost low.

Step 3: Apply Rotation Transformations

For each angle θ_j, we compute the rotated coordinates for all observations. The rotation transformation is applied to the centered coordinates:

$$x'_{ij} = \bar{x} + (x_i - \bar{x})\cos(\theta_j) - (y_i - \bar{y})\sin(\theta_j)$$
$$y'_{ij} = \bar{y} + (x_i - \bar{x})\sin(\theta_j) + (y_i - \bar{y})\cos(\theta_j)$$

This generates a set of rotated coordinate pairs (x'_{ij}, y'_{ij}) for each observation i and rotation angle j.

Step 4: Construct Augmented Feature Matrix

The final feature matrix for each observation consists of: - Original coordinates: (x, y) - Rotated coordinates: $(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_{22}, y'_{22})$

In total we have **46 features** for coordinate-rotated models (compared to 2 features for baseline models using only original coordinates).

5.4.2 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.

5.4.3 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.

This interpretation makes explicit the connection between coordinate rotation and oblique decision boundaries. An axis-aligned split applied to a rotated coordinate, such as $(x'_j < c)$, corresponds to an oblique split in the original coordinate system, which can be written as:

$$(x - \bar{x})\cos(\theta_j) - (y - \bar{y})\sin(\theta_j) < c - \bar{x}$$

This condition defines a linear decision boundary in the original space with slope tan( ). In this sense, coordinate rotation allows standard axis-aligned tree algorithms to approximate oblique splits without explicitly modifying the tree induction procedure.

5.4.4 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 € {1, …, 23}, is given by the formula 360 * (i / 23), The 23-th is the real one.

5.5 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.

Table X: Benchmark Configuration

Component	Specification	Description
Train/Test Split	80% / 20%	Training set for model fitting, test set for performance evaluation
Random Seed	42	Fixed seed for reproducible train/test splitting and model initialization
Test Size	Variable by dataset	Small datasets: 5,000 points; Large datasets: 100,000 points
Coordinate Rotation	23 angles	Preprocessing for tree-based models (when applicable)
Target Transformation	Log transform	Applied to real elevation datasets (BDALTI, RGEALTI) to handle skewness
Hardware	SSPCloud environment	Multi-core CPU, optimized for parallel computation
Software	Python 3.13	scikit-learn, XGBoost, PyKrige, pyGAM, GeoRF, Polars, scikit-learn-intelex
Parallelization	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^2 (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.

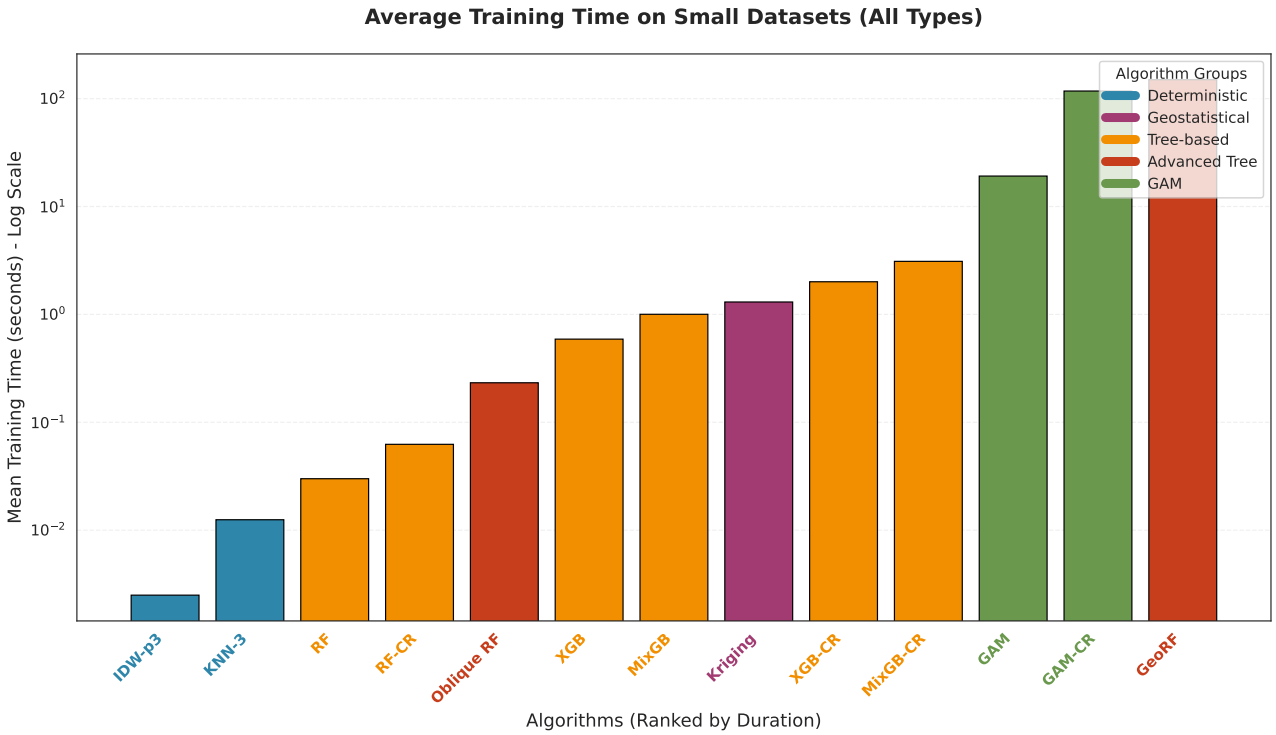
All metrics are computed on the test set to assess generalization performance on unseen data. For models requiring hyperparameters we use configurations established through preliminary tuning or standard recommendations from the literature. The focus of this benchmark is on comparing algorithm families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

6 Results

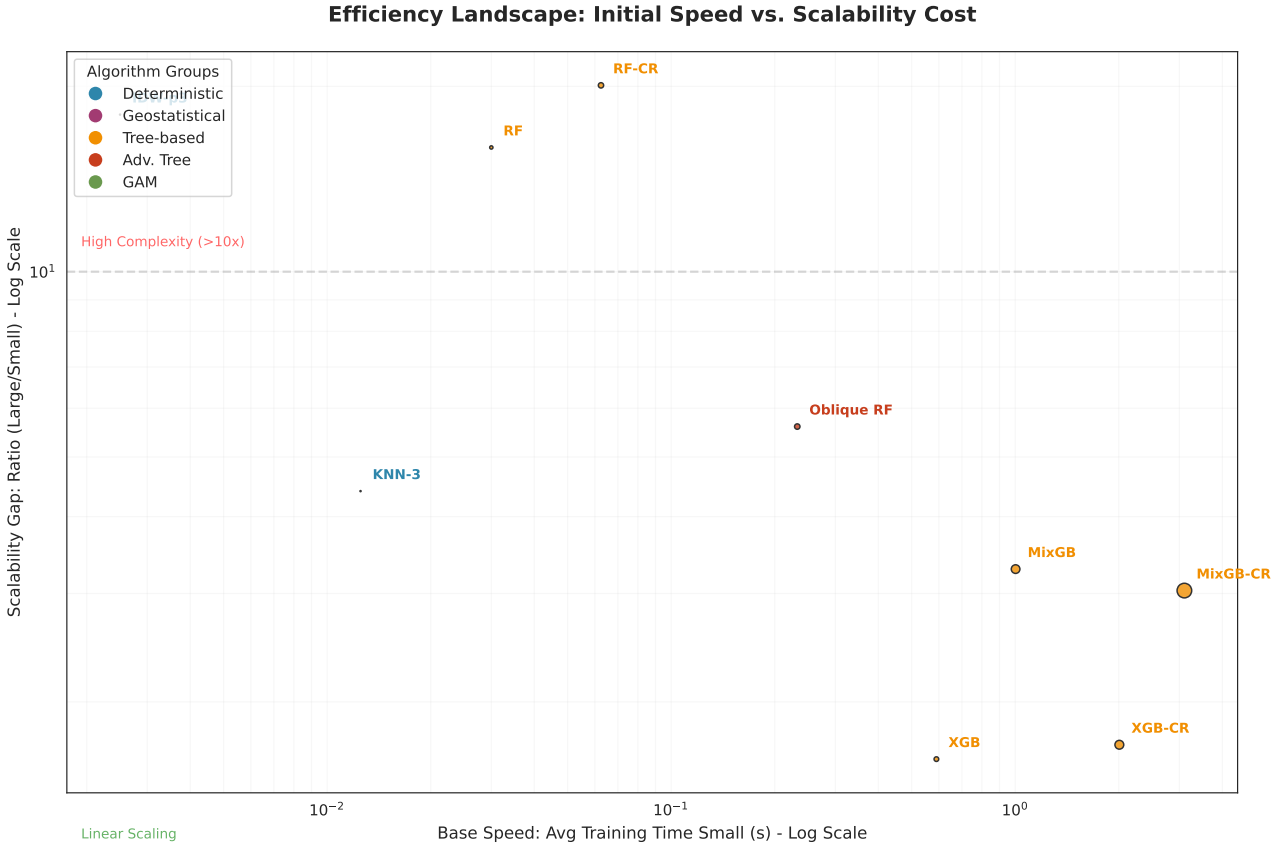
6.1 Analysis per Category

6.1.1 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.

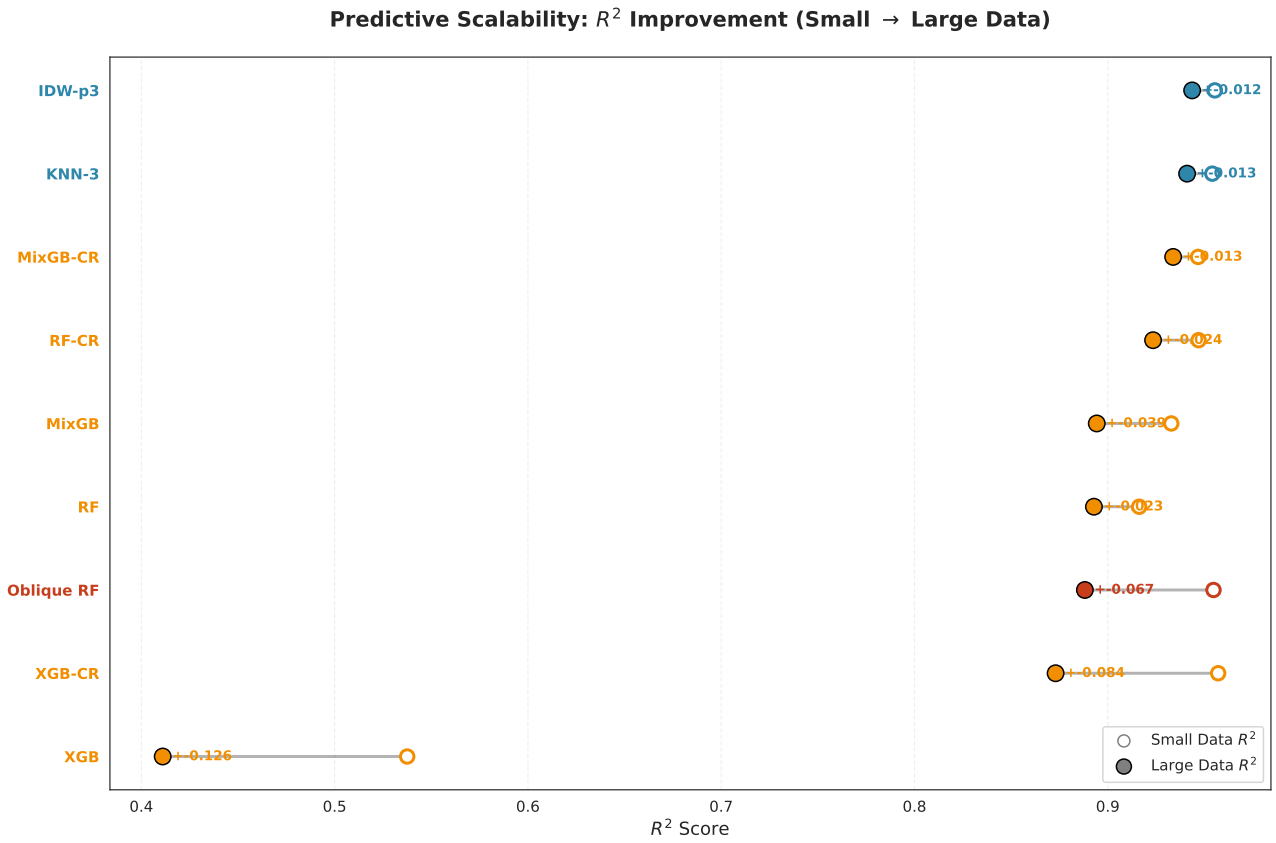


Here we can see that determnistic algorithm are the faster, on the opposite GeoRF and GAM models seems to be very slow. Based on that, we decided to not run GAM and GeoRF on big dataset and see what is happening. Then we try to run the rests of algorithms on big dataset, and we remark that krigning is 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.



6.1.2 Scalability

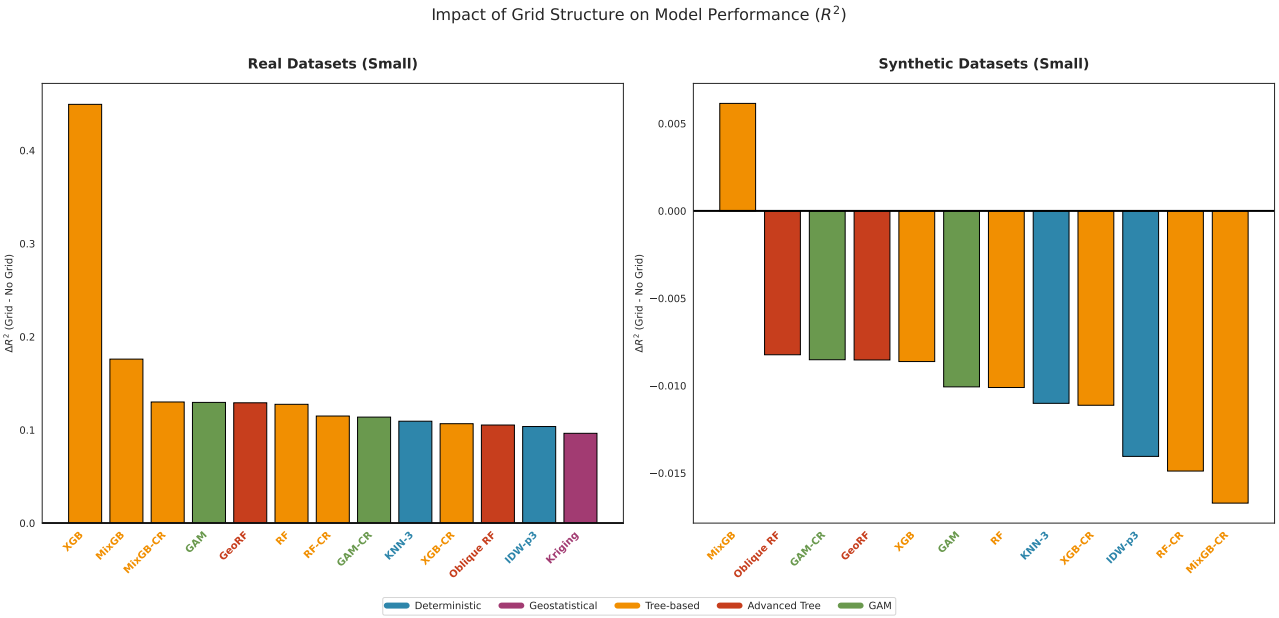
Once the efficient algorithms are identified, we examine the "Size Effect" to determine if increasing the volume of data provides a significant return on investment in terms of R2. The transition from small to large datasets generally leads to a performance jump for all models, but the magnitude varies.



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.

6.1.3 Grid or real what is relevant ?

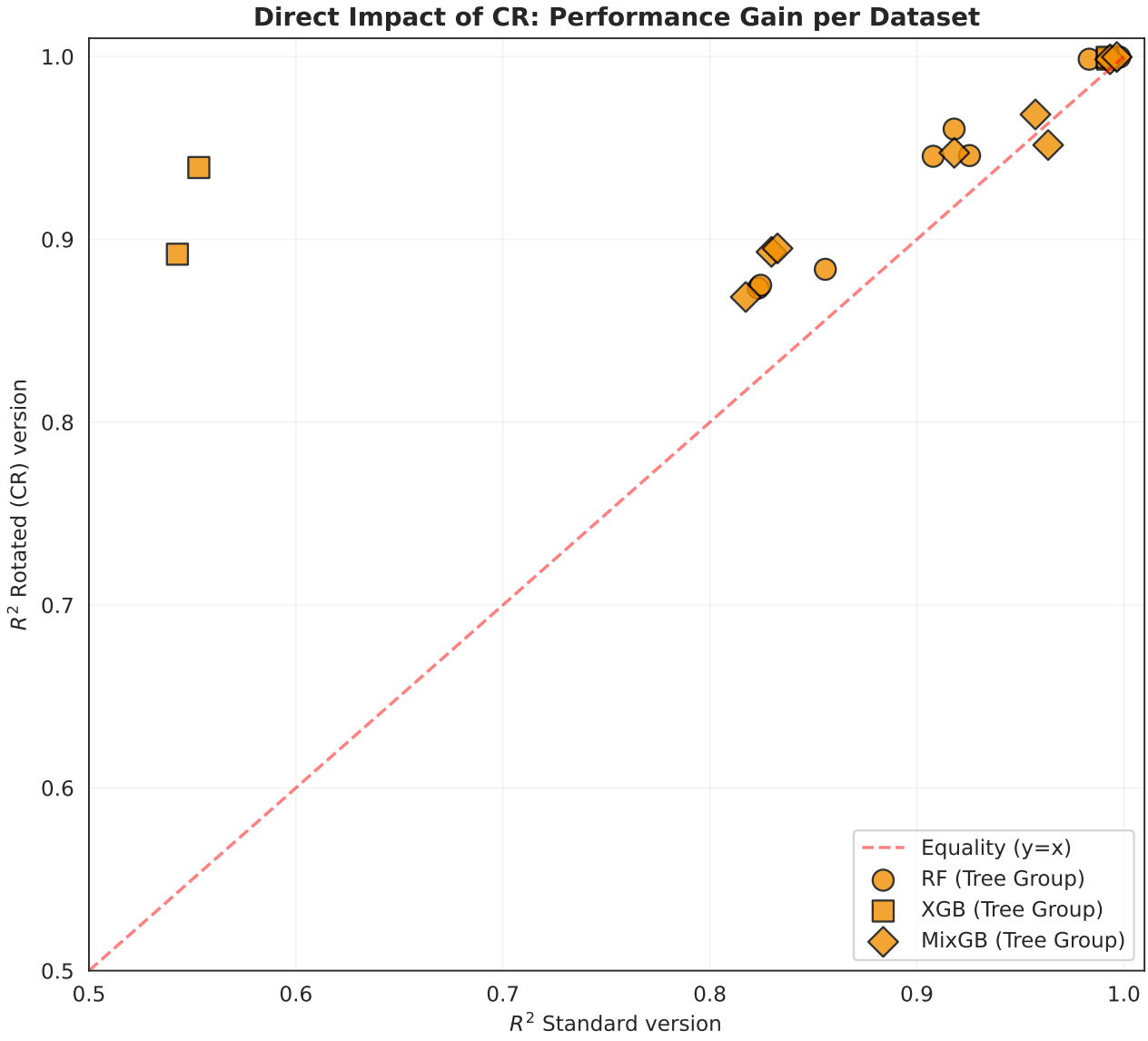
Beyond dataset size, the geometric arrangement of points plays a critical role in model performance.



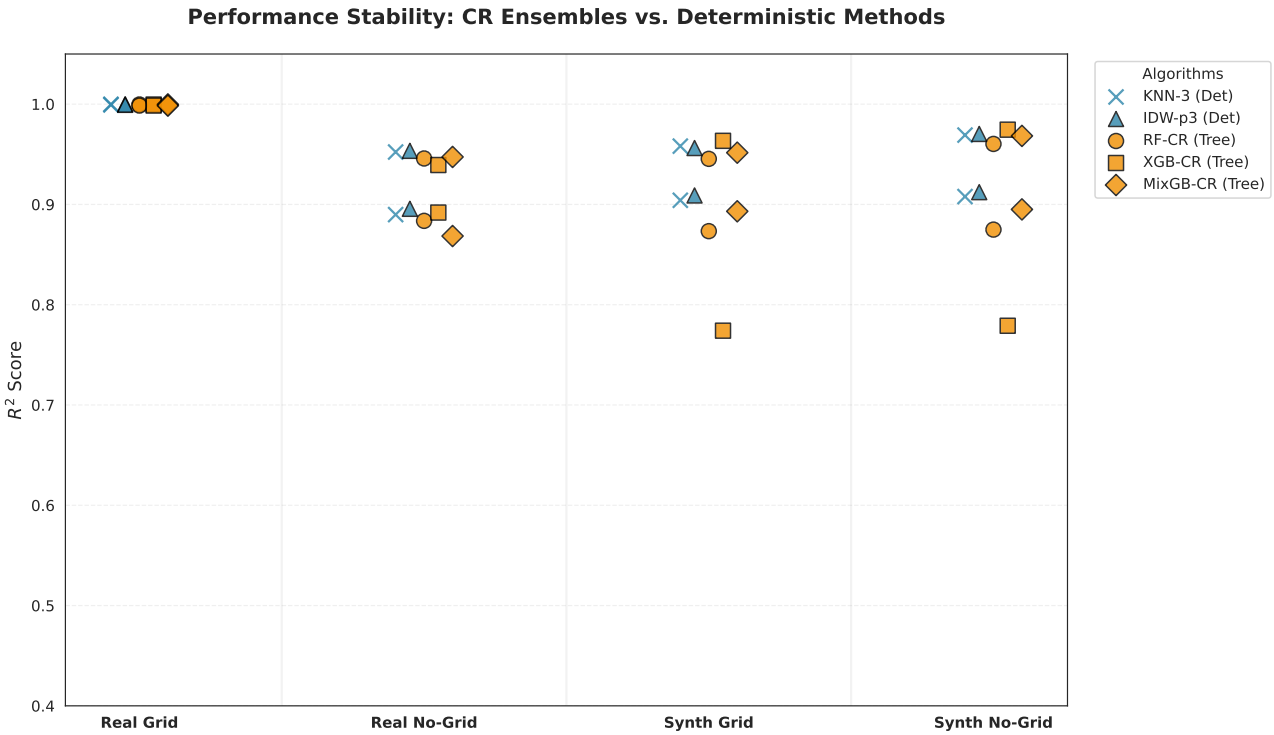
The comparison between structured grids and unstructured "no-grid" samples reveals a specific bias in tree-based algorithms toward axis-aligned structures. Similarly, the transition from synthetic stationary fields to complex real-world topography (RGE and BD ALTI) tests the robustness of the spatial 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.

6.2 Effects of Coordinate Rotation

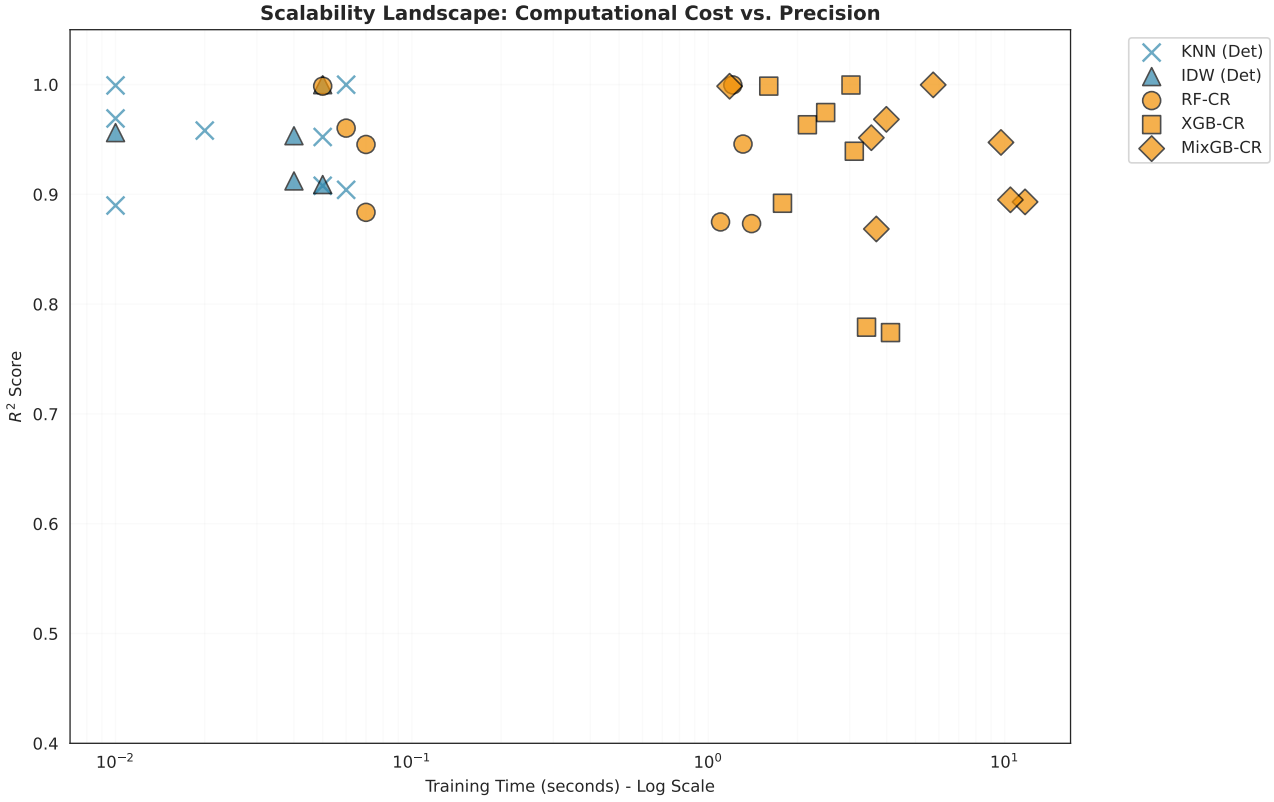
Having established the baseline behavior of the models, we turn to the primary focus of this study: the impact of Coordinate Rotation (CR). The evidence clearly indicates that CR consistently improves ensemble methods across nearly all dataset types. By projecting coordinates into a rotated space, we 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.



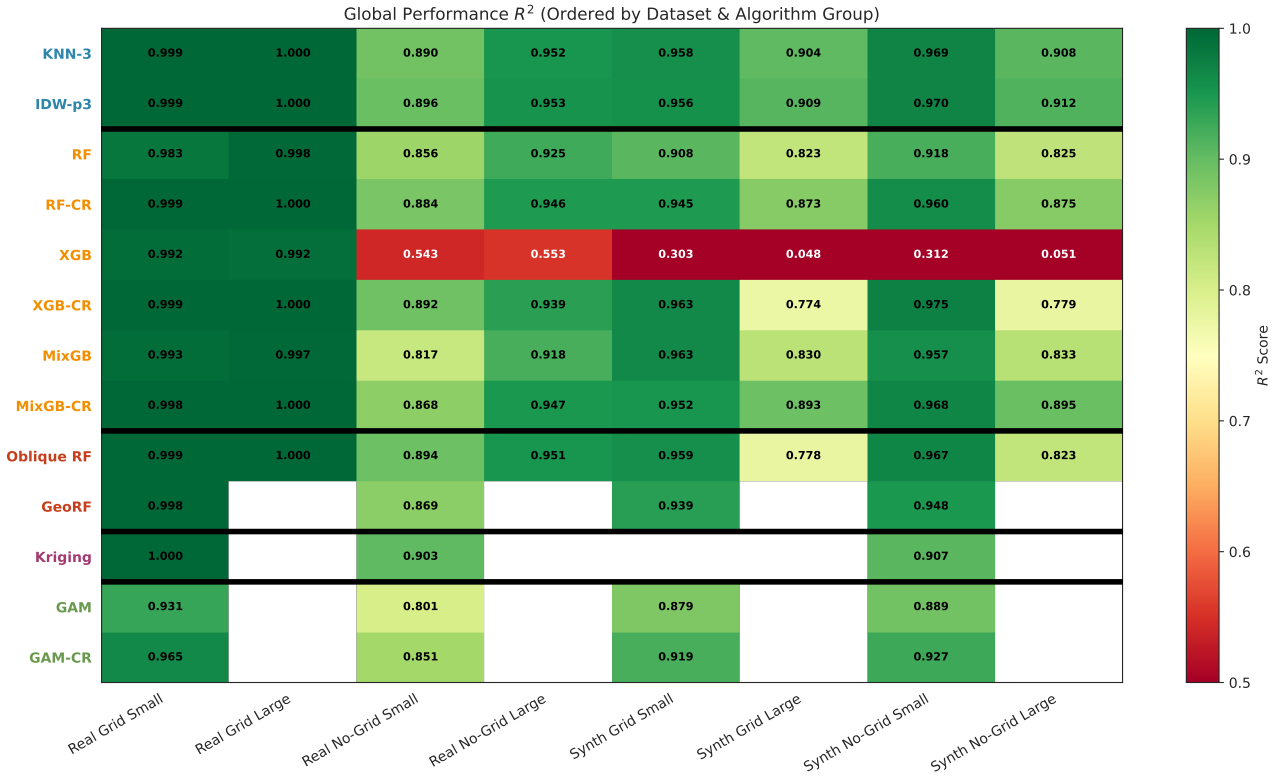
The final consideration for Coordinate Rotation is its scalability. While adding a rotation step introduces a slight overhead, the scalability graph reveals that the "time penalty" is negligible compared to the massive gain in R2. The CR-enhanced models occupy the top-right quadrant of our efficiency 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.



6.3 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 R2 scores regardless of whether the data is real, synthetic, large, or irregularly sampled.



7 Discussion

7.1 Interpretation of Results

Explanation of observed performance patterns.

7.2 Practical Implications

Guidelines for practitioners choosing interpolation methods.

7.3 Limitations

Discussion of methodological and computational limitations.

7.4 Future Work

Possible extensions and improvements.

8 Conclusion

Summary of key findings and contributions.

9 References

All references cited in the report.

10 Appendix A: Detailed Results

10.1 Synthetic Small Grid

Detailed tables and figures.

10.2 Synthetic Small No-Grid

Detailed tables and figures.

10.3 Synthetic Large Grid

Detailed tables and figures.

10.4 Synthetic Large No-Grid

Detailed tables and figures.

10.5 Real Small Datasets

Detailed tables and figures.

10.6 Real Large Datasets

Detailed tables and figures.

11 Appendix B: Algorithm Parameters

Detailed hyperparameter settings for each algorithm.