Electronic Supplementary Material (ESI) for Physical Chemistry Chemical Physics.
This journal is © the Owner Societies 2021

Determining Usefulness of Machine Learning in Materials Discovery Using Simulated Research Landscapes.

Electronic Supplementary Information

Marcos del Cueto* and Alessandro Troisi*

*corresponding authors: Marcos del Cueto, Alessandro Troisi
Email: m.del-cueto@liverpool.ac.uk, a.troisi@liverpool.ac.uk

Contents:

S1. Pseudo-code of exploration algorithms
- **S1.a.** N_1 *a*-weight exploration
- **S1.b.** N_2 *a*-weight exploration
- **S1.c.** N_2 ML-guided exploration

S2. Estimation of *a* and *S* from arbitrary datasets
S3. *a*-bias effect on RMSE with different dimensionality and smoothness
S4. *a*-bias effect on RMSE with different ML algorithms
- **S4.a.** 10-fold cross-validation
- **S4.b.** 10-fold cross-validation *vs* last-10% validation

S5. ML-guided exploration prediction error
S6. Direct comparison of *a*-weighted and ML-guided explorations
S7. ML_{gain} for other dimensionalities

S1. Pseudo-code of exploration algorithms

S1.a. N_1 *a*-weight exploration

- Explore N_0 random initial configurations as a starting database: $\{\boldsymbol{x}, G(\boldsymbol{x})\}$
- For N_1 steps:
  - **i.** Rank dataset by their $G(\boldsymbol{x})$ value
  - **ii.** Randomly choose one configuration $\boldsymbol{x}''$ from the top $a\%$ of ranked dataset
  - **iii.** Record one new $(\boldsymbol{x}', \boldsymbol{G}(\boldsymbol{x}'))$configuration within a threshold $\Delta\boldsymbol{x}$ of $\boldsymbol{x}''$
  - **iv.** Add new$(\boldsymbol{x}', \boldsymbol{G}(\boldsymbol{x}'))$ to dataset

S1.b. N_2 *a*-weight exploration

- Use the dataset after N_1 a-weight exploration as a starting dataset: $\{\boldsymbol{x}, G(\boldsymbol{x})\}$
- For N_2 steps:
  - **i.** Rank dataset by their $G(\boldsymbol{x})$ value
  - **ii.** Randomly choose one configuration $\boldsymbol{x}''$ from the top $a\%$ of ranked dataset
  - **iii.** Record one new $(\boldsymbol{x}', \boldsymbol{G}(\boldsymbol{x}'))$ configuration within a threshold $\Delta\boldsymbol{x}$ of $\boldsymbol{x}''$
  - **iv.** Add new $(\boldsymbol{x}', \boldsymbol{G}(\boldsymbol{x}'))$ to dataset

S1.c. N_2 ML-guided exploration

- Use the dataset after N_1 a-weight exploration as a starting dataset: $\{\boldsymbol{x}, G(\boldsymbol{x})\}$
- For N_2 steps:
  - **i.** Train ML model with all $\{\boldsymbol{x}, G(\boldsymbol{x})\}$ in dataset
  - **ii.** Predict $G(\boldsymbol{x}')$ for configurations within a threshold $\Delta\boldsymbol{x}$ of all previous $\boldsymbol{x}$ in dataset
  - **iii.** Add to dataset the $(\boldsymbol{x}', \boldsymbol{G}(\boldsymbol{x}'))$ point with minimum $G(\boldsymbol{x}')$

The complete code, along with instructions and examples, can be found in this online repository: https://github.com/marcosdelcueto/MachineLearningLandscapes

S2. Estimation of *a* and *S* from arbitrary datasets

Given the definition we introduced of adventurousness: future points are obtained by small variations of points in the top *a* percentile of previous points, we can estimate the adventurousness of a time-ordered data set following the general procedure:

- For a given set of Δa values, consider a series of time-ordered data points $\mathbf{x}_i$.
- For each of the *i* points, calculate what is the d_{min} distance of $\mathbf{x}_i$ with the closest $\mathbf{x}_j$ data point in the *a* percentile, such that $j<i$.
- Then, get the maximum d_{min} distance of all considered points, $d_{max\text{-}min}$.
- Finally, we can represent $d_{max\text{-}min}$ as a function of *a*. $d_{max\text{-}min}$ measures what the largest closest-distance of a point *i* with the previous points in the top *a* percentile is. $d_{max\text{-}min}$ will decrease as *a* increases, until it levels off, indicating what the estimated *a* is.

We show the *a* estimation using this methodology in Figure S1, averaged over 100 datasets with a specific (n,N_1,a,S) combination.

Our definition of smoothness directly influences the corrugation of the data. Thus, one can estimate *S* following:

- Calculate the ΔG and $\Delta\boldsymbol{x}$ differences of each point *i* in the data set with all other *j* points.
- Calculate corrugation as the mean value of $\Delta G/\Delta\boldsymbol{x}$ for all points.
- Estimate *S* as the weighted inverse of the corrugation. This weight allows us to adjust *S* values to our arbitrary $(0,1/3)$ range.

We also calculated the *S* estimation for 100 research landscapes with a specific (n,N_1,a,S) combination in Figure S2. These cases intend to serve as an example of how one can successfully approximate *a* and *S* to take full advantage of the guiding capabilities of ML_{gain}.

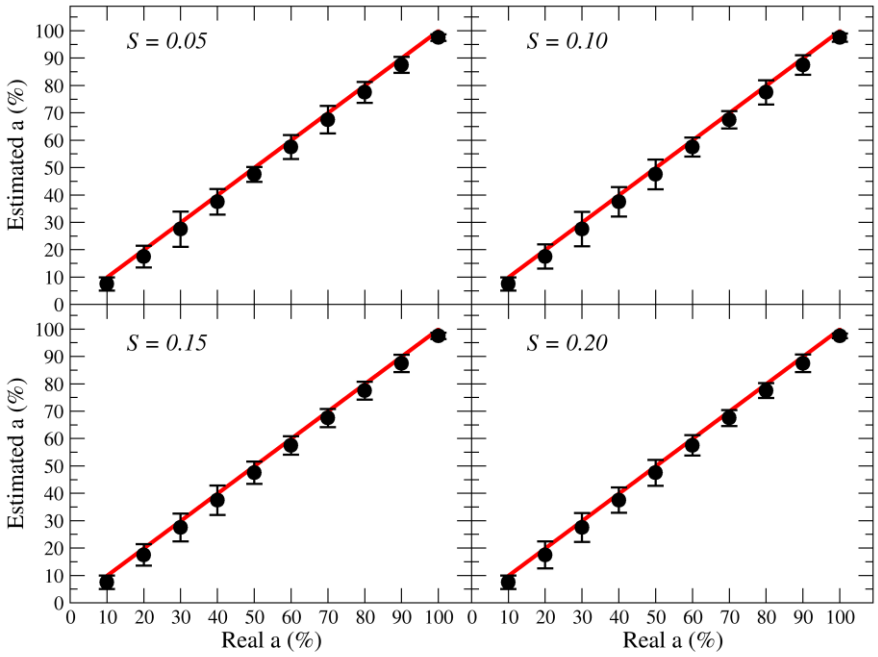

Fig. S1. Estimated *a* values for *n=3*, N_1*=200*. We represent the median value, plus/minus standard deviation of the estimation of 100 datasets. Red lines indicate what the value of an ideal estimation would be.

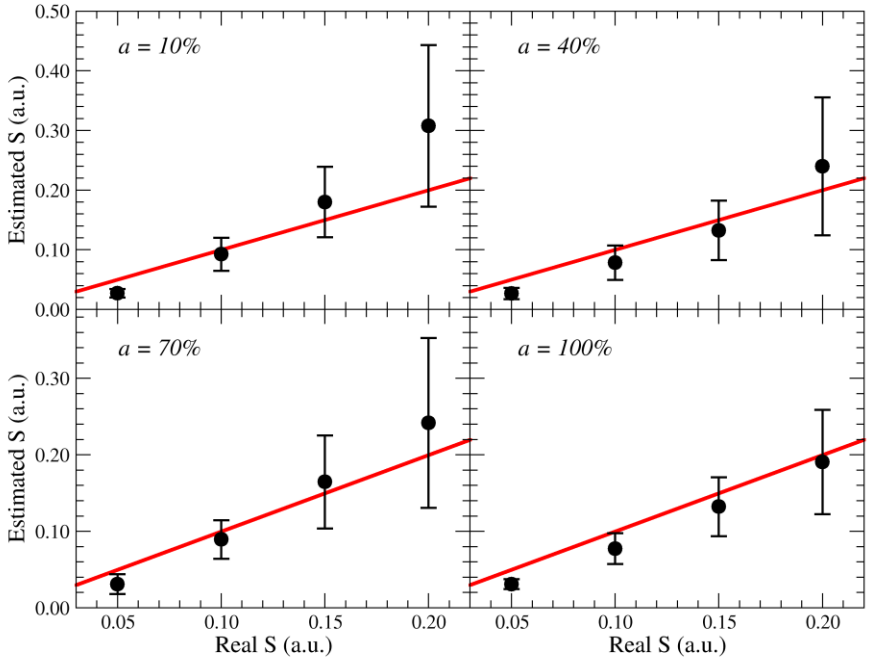

Fig. S2. Estimated *S* values for *n=3*, N_1*=200*. We represent the median value, plus/minus standard deviation of the estimation of 100 datasets. Red lines indicate what the value of an ideal estimation would be.

S3. *a*-bias effect on RMSE with different dimensionality and smoothness

In Figures S3-S4, we show the RMSE values for different smoothness and dimensionalities, using GPR (for *n=3*, see Figure S8). We observe the same trends in all cases, where the *a*-bias (RMSE difference between *a=10%* and *a=100%*) tends to decrease when increasing the dataset size or smoothness. Datasets of larger dimensionality are more complex, so the *a*-bias is still significant at N_1*=1000*, and we would need larger datasets to start to reduce the *a*-bias significantly.

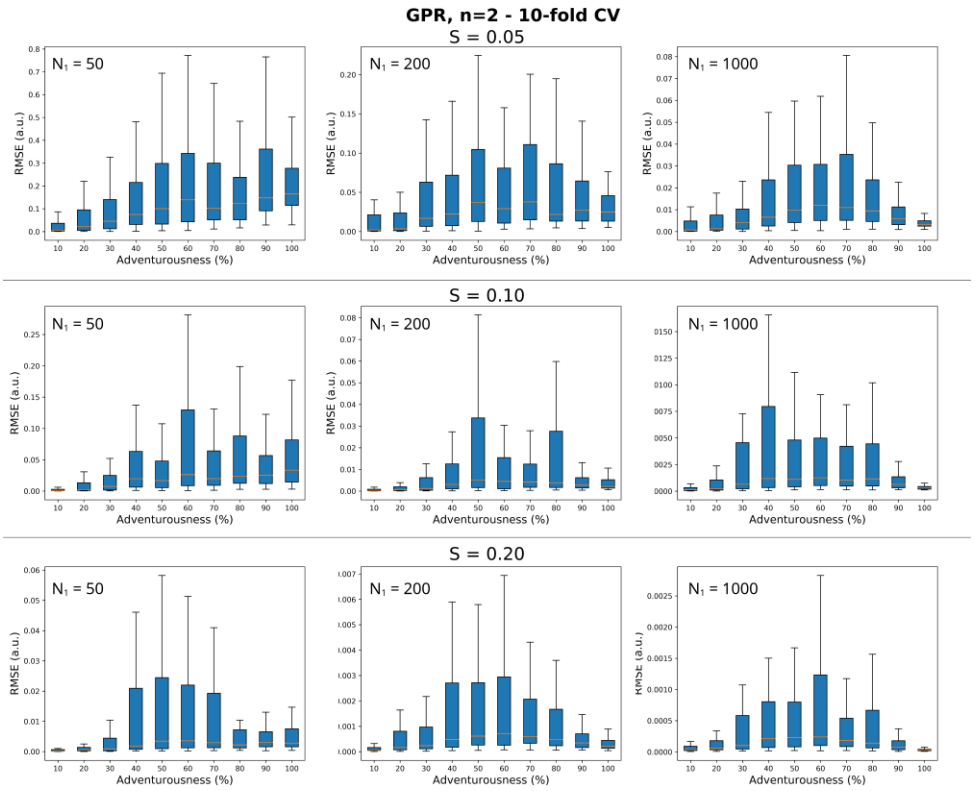

Fig. S3. Prediction error (RMSE) obtained with GPR and a 10-fold cross-validation, for different smoothness (*S*), adventurousness (*a*) and data set sizes (N_1). Dataset with dimensionality *n=2*.

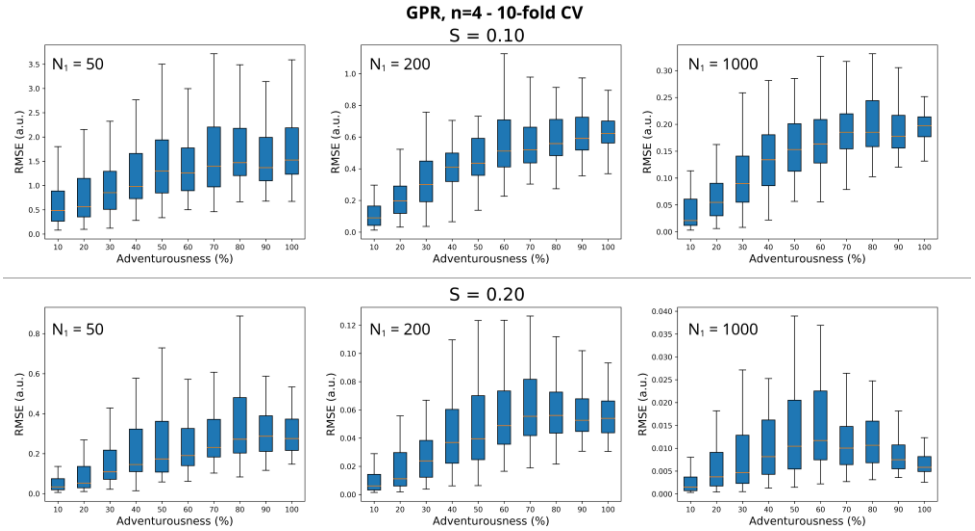

Fig. S4. Prediction error (RMSE) obtained with GPR and a 10-fold cross-validation, for different smoothness (S), adventurousness (a) and data set sizes (N_1). Dataset with dimensionality n=4.

S4. *a*-bias effect on RMSE with different ML algorithms

We have considered four ML algorithms to fit our data: i) k-Nearest Neighbours (k-NN)(1), ii) Gradient Boosting Regression (GBR)(2), iii) Kernel Ridge Regression (KRR)(3, 4) and iv) Gaussian Process Regression (GPR)(5). These methods are representative of different categories of supervised learning methods frequently used to analyze problems in materials science(6).

In k-NN, the value of a given configuration $\boldsymbol{x}$ is obtained as the mean value of its k nearest neighbors, with weights that are proportional to the Euclidean distance between $\boldsymbol{x}$ and its nearest neighbors. The optimum number of nearest neighbors k is selected in each case from a list of possible values as the one that minimizes the RMSE of the fit.

GBR is composed of an ensemble of decision trees, built in a forward stage-wise fashion to minimize a loss function. It uses a specific learning rate value to correct for mistakes on the previous stage. We use the mean square error with the improvement score by Friedman(2) as a loss function. The total number of boosting stages, learning rate, maximum depth of individual regression estimators, the minimum number of samples to split an internal node and required to be at a leaf node are all optimized from a set of possible values.

GPR makes use of weighted averages of training data, with probabilistic weights. GPR uses covariance functions (kernels) to measure the similarity between points. In our case, we have used an RBF kernel:

$$k(x_i, x_j) = e^{-\gamma|x_i - x_j|^2}$$

where γ is the Gaussian kernel variance. The variance of the estimation is further controlled by the regularisation constant α. Both hyper-parameters, α and γ, are optimized by minimizing the RMSE of the test set with a differential evolution algorithm (7), as implemented in SciPy (8).

KRR is a generalized version of the least square procedure, with the addition of non-linearity and regularisation. We have used the same kernel that we did for GPR, whose hyper-parameters are optimized with the same differential evolution algorithm(7).

S4.a. 10-fold cross-validation

To train these algorithms, we have used a 10-fold cross-validation, in which the original sample is randomly partitioned into ten equal-sized subsamples. Of the ten subsamples, a single subsample is retained as the validation data for testing the model, and the remaining nine subsamples are used as training data. The cross-validation process is repeated ten times, with each of the ten subsamples used exactly once as the validation data. The final RMSE is calculated by comparing all predicted and actual values at each fold. Hyperparameters are optimized to minimize this RMSE. This cross-validation method is standard for advanced properties data sets in material science, containing typically between hundreds and thousands of entries(6).

We show in Figures S5-S8 how the RMSE resulting with this cross-validation, using different smoothness and ML algorithms, for *n=3*. We observe in all cases that RMSE decreases when increasing N_1. Methods more sophisticated, as KRR and GPR produce smaller RMSE values, and we can see how they are able to overcome the a-bias when the research landscapes are large and smooth. GPR was finally chosen as the example to show in the main manuscript, as the results are slightly better with GPR overall.

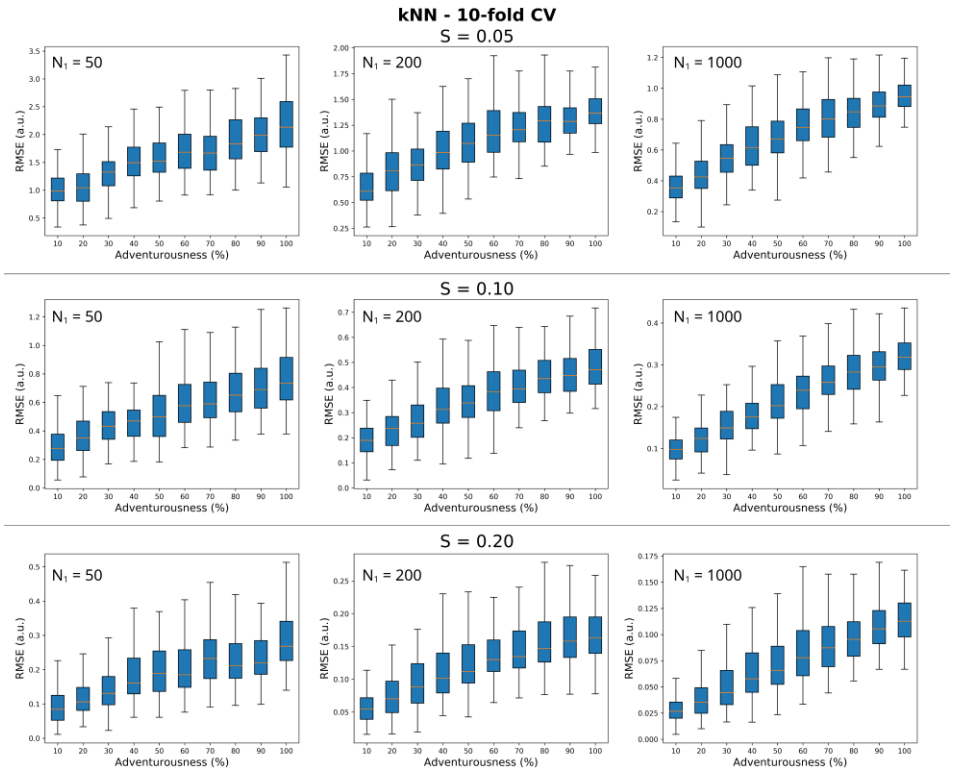

Fig. S5. Prediction error (RMSE) obtained with k-NN and a 10-fold cross-validation, for different smoothness (*S*), adventurousness (*a*) and data set sizes (N_1). Dataset with dimensionality $n=3$.

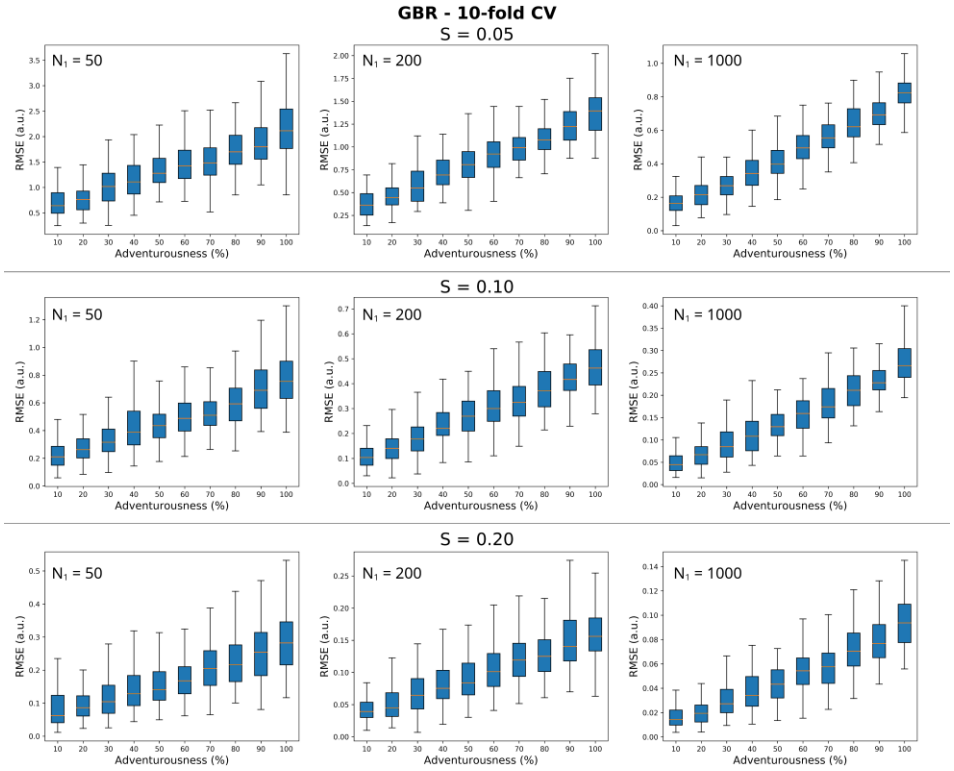

Fig. S6. Prediction error (RMSE) obtained with GBR and a 10-fold cross-validation, for different smoothness (*S*), adventurousness (*a*) and data set sizes (N_1). Dataset with dimensionality $n=3$.

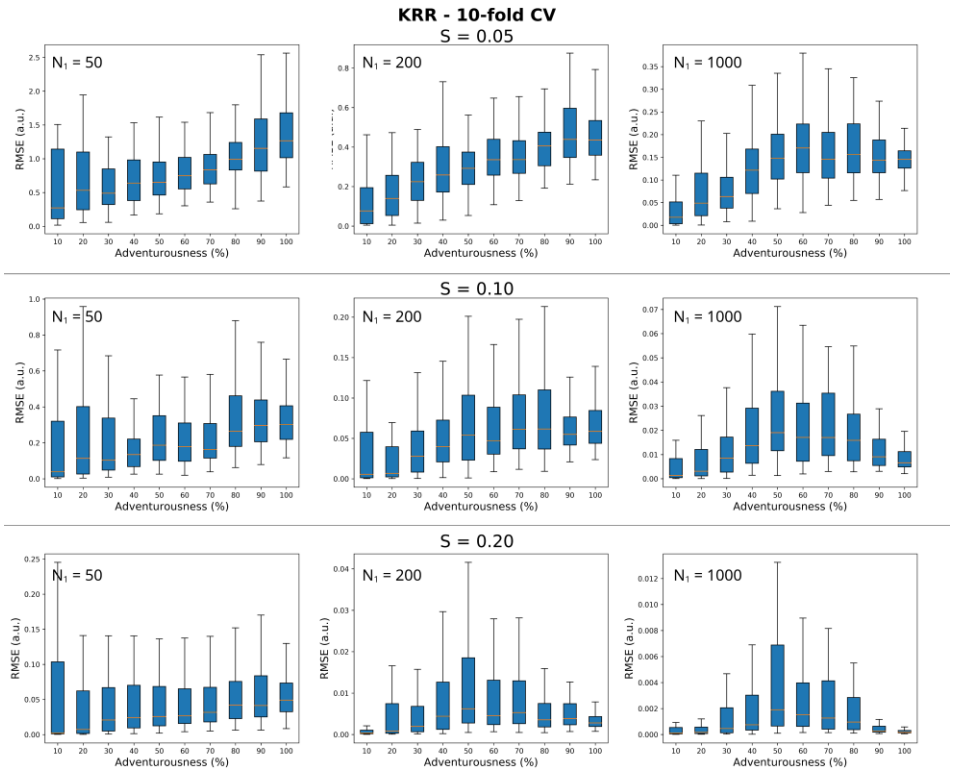

Fig. S7. Prediction error (RMSE) obtained with KRR and a 10-fold cross-validation, for different smoothness (*S*), adventurousness (*a*) and data set sizes (N_1). Dataset with dimensionality $n=3$.

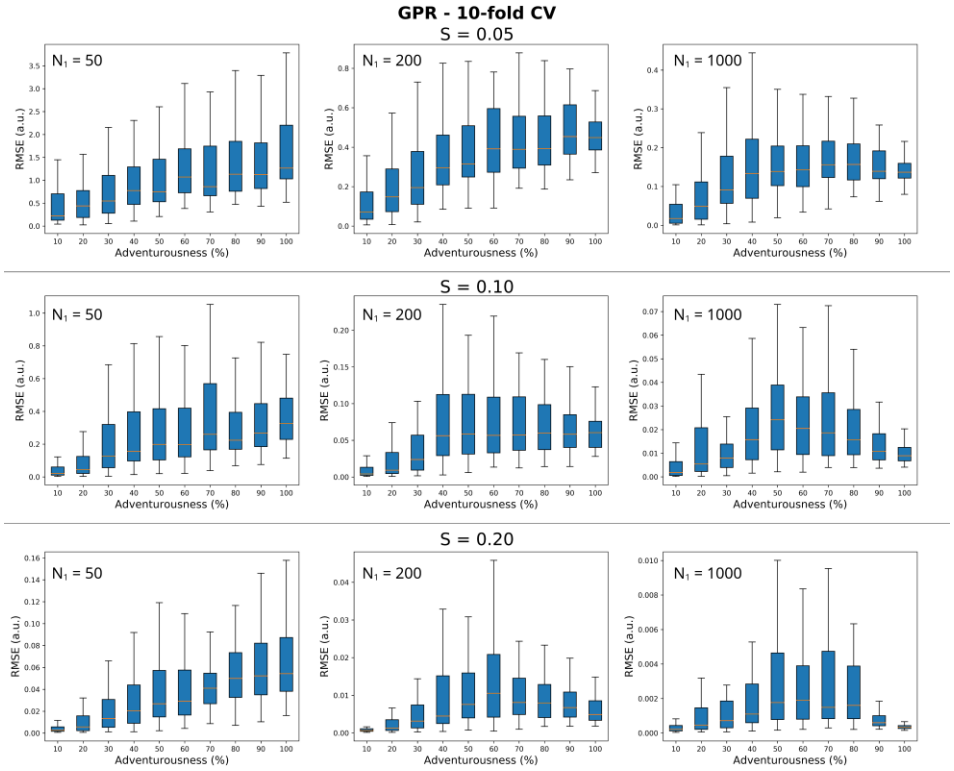

Fig. S8. Prediction error (RMSE) obtained with GPR and a 10-fold cross-validation, for different smoothness (*S*), adventurousness (*a*) and data set sizes (N_1). Dataset with dimensionality $n=3$.

S4.b. 10-fold cross-validation *vs* last-10% validation

In Figure S9, we show how the *a*-bias changes with the different validation methods, for the four ML algorithms studied. With a last-10% validation, we use a 10-fold cross-validation within the first 90% of the time-ordered dataset to optimize the hyperparameters, and report the resulting RMSE for the last 10% of the time-ordered dataset.

All algorithms display similar trends. It is observed how the *a*-bias (RMSE difference between *a=10%* and *a=100%*) increases when using a last-10% validation, which means that the data-bias introduced in the ML model is amplified when the dataset is time-ordered.

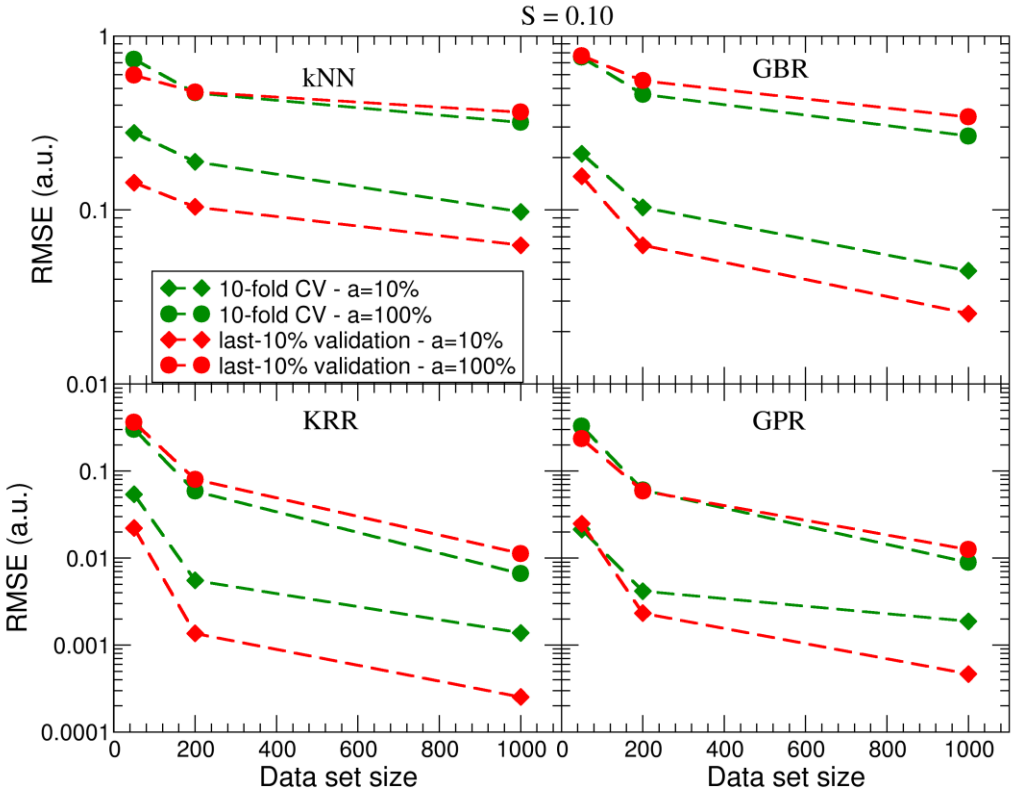

Fig. S9. Prediction error (RMSE) obtained with a 10-fold cross-validation (green) and a last-10% validation (red), for *a=10%* (diamonds) and *a=100%* (circles), for different ML algorithms and dataset sizes. Datasets of dimensionality *n=3*.

S5. ML-guided exploration prediction error

In Figure S10, we show the relative error committed during the ML-guided exploration. To calculate this error, we compare the minimum predicted *G* value and the real value at that configuration. This error would be, in principle, unknown in real instances until an experiment is performed at that configuration. However, the use of research landscapes allows us to know what this error would be. The error is calculated with the following equation:

$$\varepsilon_{ML} = \left| \frac{G_{min}^{ML-predicted} - G_{min}^{ML-real}}{G_{min}^{ML-real}} \right|$$

We can see in this figure how the prediction error is only significant when we are working with small datasets (N_1<100), but it rapidly becomes negligible as one considers larger datasets.

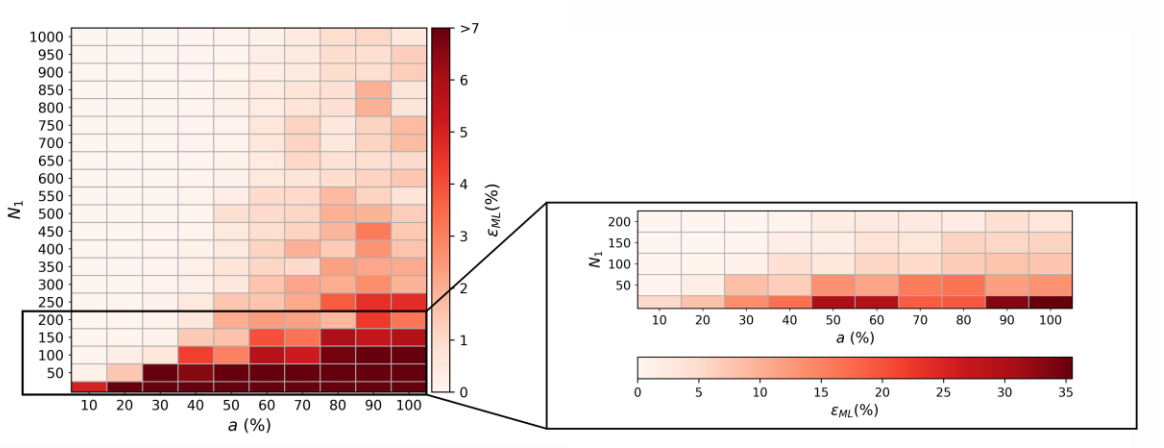

Fig. S10. ML-guided exploration relative prediction error for different data set sizes (N_1) and adventurousness (a) obtained for datasets with a dimensionality $n=3$.

S6. Direct comparison *a*-weighted and ML-guided explorations

After generating the research landscapes of size N_1 with an *a*-weighted exploration algorithm, we perform two explorations for $N_2=15$ steps, using i) *a*-weighted exploration, and ii) ML-guided exploration. In Figure S11, we show the comparison of the minimum *G* values (G_{min}) found during the *a*-weighted explorations (in black circles) and the ML-guided exploration (in red diamonds) for a few *a* significant values. Note that both G_{min} values are very similar for lower adventurousness, but their difference starts to increase significantly as adventurousness increases. The relative difference between these G_{min} values will eventually be used to calculate ML_{gain}.

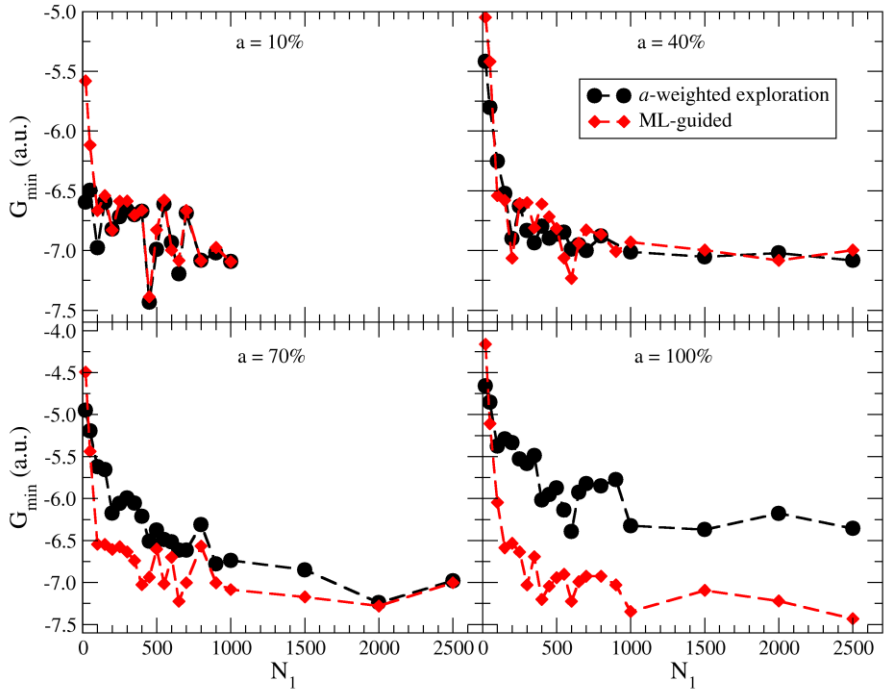

Fig. S11. Minimum *G* value obtained with GPR an $n=3$ with different data set sizes (N_1) and adventurousness (a) with an *a-weighted* exploration (black) and ML-guided exploration (red).

S7. ML_{gain} for other dimensionalities

In Figure S12, we show how the ML_{gain} values change for different *a* and *n* values. With all these values, we observe the same trend, where a window of opportunity to maximize ML_{gain} is observed. We observe how ML_{gain} is negative or very close to zero for small datasets, and it becomes negligible again when one uses very large datasets. It is for intermediate data set sizes, and large *a* values, that one sees a significant ML_{gain}. ML_{gain} tends to increase with dimensionality, which is not surprising, since there is more room for discovering new best-performing materials as one increases the underlying complexity of the field.

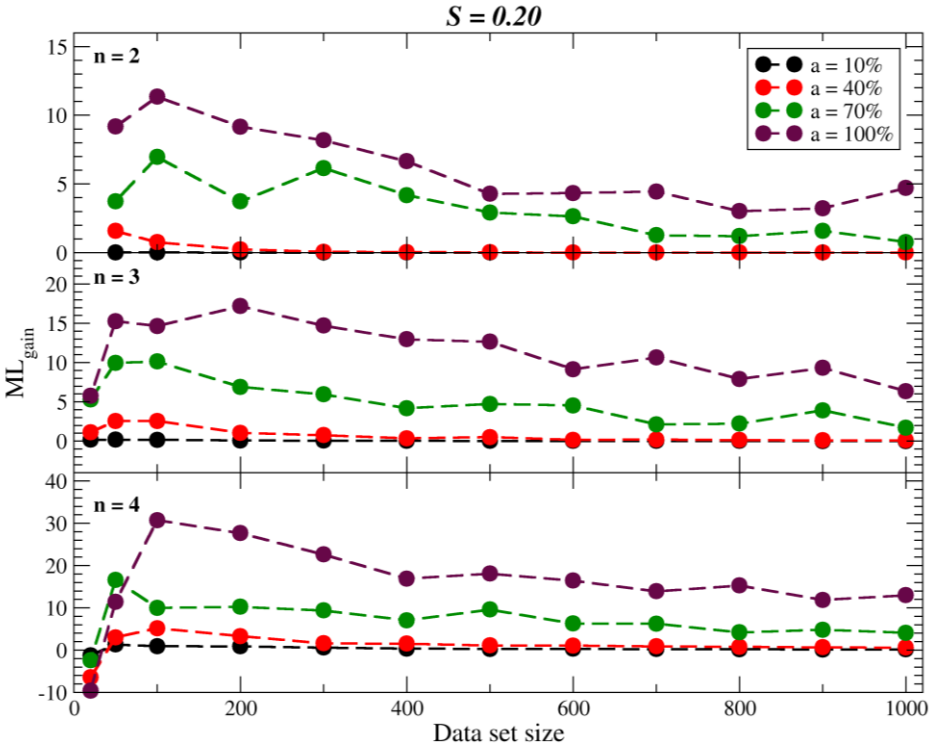

Fig. S12. : Different 1D cuts of ML_{gain} for different dataset sizes and dimensionalities (*n*), for a few significant adventurousness (*a*) values.

References

1. N. S. Altman, An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician* **46**, 175–185 (1992).
2. J. H. Friedman, Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* **29**, 1189–1232 (2001).
3. K. Vu, *et al.*, Understanding kernel ridge regression: Common behaviors from simple functions to density functionals. *International Journal of Quantum Chemistry* **115**, 1115–1128 (2015).
4. M. Rupp, Machine learning for quantum mechanics in a nutshell. *International Journal of Quantum Chemistry* **115**, 1058–1073 (2015).
5. E. O. Pyzer-Knapp, G. N. Simm, A. A. Guzik, A Bayesian approach to calibrating high-throughput virtual screening results and application to organic photovoltaic materials. *Mater. Horizons* **3**, 226–233 (2016).
6. C. Chen, *et al.*, A Critical Review of Machine Learning of Energy Materials. *Advanced Energy Materials* **10**, 1903242 (2020).
7. R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization* **11**, 341–359 (1997).
8. P. Virtanen, *et al.*, SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* **17**, 261–272 (2020).