



Cite this: DOI: 10.1039/d5dd00565e

Automatic generation of input files with optimised k -point meshes for Quantum ESPRESSO self-consistent field single-point total energy calculations

Elena Patyukova,^{*a} Junwen Yin,^{ID} ^{*a} Susmita Basak,^b Samuel Pinilla,^{ID} ^{bc}
Alin M. Elena^a and Gilberto Teobaldi^{ID} ^b

Performing density functional theory (DFT) calculations requires a careful choice of computational parameters to ensure convergence and obtain meaningful results. This represents a particularly important problem for high-throughput and agentic workflows, where due to computational cost, any additional convergence studies are preferably avoided. So, there is a need for tools and models which are able to predict DFT parameters from basic input information, such as a structure. In this work, we develop a machine learning approach to predict the appropriate k -point sampling in DFT calculations and generate the input files for Quantum Espresso self-consistent field calculations. To achieve this, we first generated a training dataset comprising over 20 000 materials, each with an energy convergence threshold of 1 meV per atom. Several ML models were evaluated for their ability to predict k -point distance, and uncertainty estimation was incorporated to guarantee that, for at least 85–95% of compounds, the predicted k -distance lies within the convergence region. The best-performing models are made publicly available through an open-access web application.

Received 16th December 2025

Accepted 31st May 2026

DOI: 10.1039/d5dd00565e

rsc.li/digitaldiscovery

1 Introduction

High-throughput calculations in the age of AI are an invaluable source of data that drive materials discovery pipelines and the development of new models and algorithms. Performing high-throughput DFT calculations requires the appropriate choice of calculation parameters. In this paper, we focus on the most common type of (periodic boundary and plane wave) density functional theory (DFT) calculations—the self-consistent field (SCF) calculation, which involves choosing an appropriate software package (based on the material system and targeted properties), selecting a suitable exchange–correlation functional and pseudopotentials, defining a sufficiently dense k -point mesh, applying an appropriate electronic smearing scheme and Gaussian broadening width, and setting suitable plane wave cutoff energies for the wavefunctions and charge density, among other parameters.

Choosing the right parameters is crucial for achieving accurate results while optimizing computational resources, as improper settings can lead to convergence issues or unnecessarily long computation times. However, choosing the optimal

strategy for parameter choice is a non-trivial task, as, in general, the given level of accuracy in a property can be achieved with multiple sets of parameter combinations; the effects of different parameters are often interdependent and not well understood.

One of the most common approaches to parameter selection in high-throughput calculations is to fix all relevant parameters at certain values for all compounds, usually at sufficiently large k -point densities and energy/density cutoffs. The situation is further complicated by the fact that the rationale behind the chosen fixed parameter values is often not reported. In this situation, there is little control or understanding of the associated errors, and calculations often turn out to be partially over-converged and partially under-converged.

In recent years, there has been great interest and progress in automating high-throughput workflows both through rigid-logic automation systems such as AiiDA,¹ Atomate2,² Pyiron,³ and AFLOW⁴ and new emerging agentic LLM-driven frameworks.^{5–7} Fixed workflow systems provide robust workflows, data management tools, HPC-interfaces, input/output processing and analysis, and help standardise DFT simulations, result reproducibility, and reuse. Emerging agentic frameworks use more flexible approaches and aim at performing the full cycle of calculations and data analysis in an automated manner. Potentially, these frameworks can give non-experts and robots wide access to theoretical tools. However,

^aScientific Computing Department, Daresbury Laboratory, STFC, UKRI, UK. E-mail: patyukova@gmail.com; junwen.yin@stfc.ac.uk

^bScientific Computing Department, Rutherford Appleton Laboratory, STFC, UKRI, UK
^cDiamond Light Source, Harwell Science and Innovation Park, Didcot OX11 0DE, UK



the choice of calculation parameters for those workflows remains a problem in both cases.

In this work, we aim to improve on the conventional baseline of fixed parameters for single-point SCF energy calculations conducted using Quantum ESPRESSO⁸ by developing a tool using machine learning to predict optimal calculation parameters. By sharing this tool, we aim to help to improve the quality of high-throughput datasets and make calculations greener by reducing unnecessary calculation time.

We begin by reviewing prior work in the field. Based on this analysis and our own convergence studies, we identify an optimal strategy for data generation and subsequently construct the dataset. We then benchmark different models to determine which achieves the lowest prediction errors for k -point density. A key consideration in this task is uncertainty estimation, since the cost of prediction errors is asymmetric: underestimating the required k -point density can lead to inaccurate results, while overestimating it only increases computational expense. To address this, we predict the 0.85, 0.9, and 0.95 quantiles rather than the median (0.5 quantile), ensuring that at least 85%, 90%, and 95% of our predictions are not underestimated.

Finally, we encapsulate our findings in a practical web application that automatically generates input files for single-point SCF calculations in Quantum ESPRESSO for any given structure.

1.1 Contributions of this work

In this paper, we make the following contributions:

1. A large-scale convergence dataset of 20 178 compounds, generated with consistent SSSP-1.3 Efficiency pseudopotentials and cold smearing, providing reliable reference k -meshes for SCF single-point energy calculation, with a convergence criterion of 1 meV per atom.
2. A systematic evaluation of machine learning models (Random Forest, Gradient Boosting, CrabNet, CGCNN, and ALIGNN) and feature types (composition, structure, SOAP, lattice descriptors, and metallicity embeddings) for predicting optimal k -point densities.
3. A compact and interpretable rule-set derived *via* surrogate decision-tree modeling, offering intuitive insight into the key physical and chemical factors influencing k -point spacing.
4. An uncertainty estimation framework using conformalised quantile regression (CQR), ensuring that predictions satisfy probabilistic guarantees and avoid underestimating required k -point density.
5. A publicly available web application that automatically generates fully specified Quantum ESPRESSO SCF input files using the trained models.
6. Quantitative demonstration of computational savings, showing that ML-driven predictions reduce unnecessary k -mesh density and thereby lower computation time relative to fixed-density baselines.

1.2 Previous work

The most common approach to the determination of parameters for the DFT calculations is to perform a convergence curve

modeling with respect to different parameters, such as k -point mesh, cutoff energies, and smearing. However, in the case of high-throughput calculations, this is in general unreasonable due to the amount of calculation time required. So, fixing those parameters for all compounds in the high-throughput dataset at some constant chosen by an expert was adopted. There were multiple attempts to improve the quality of this choice. Choudhary and Tavazza⁹ developed an automatic convergence procedure to derive k -point densities and energy cut-off values to reach a 1 meV per cell convergence in energy for $\sim 30\,000$ materials, using the Vienna *ab initio* Simulation Package (VASP). The authors used projector-augmented wave (PAW) pseudopotentials and Gaussian smearing (with a fixed value of 0.01 eV) for all calculations. The authors tried to find correlations between k -points, cut-off parameters, and other material properties such as density, number of electrons, the slope of bands, number of band-crossings, the maximum plane-wave cutoff used for pseudopotential generation, crystal system, the number of unique species in the material, and exchange-correlation functional. They discovered that: (1) convergence is nearly independent of the exchange-correlation functional; (2) the converged plane wave cutoff is significantly correlated with the maximum cutoff used in pseudopotential generation (ENMAX) (Pearson's correlation ≈ 0.64); (3) strong correlation exists between the slope of a band crossing at the Fermi surface for metals and the required k -point density (Pearson's correlation ≈ 0.66); (4) certain preferences exist for using higher k -points (often in cubic and hexagonal) *vs.* higher plane-wave cut-off energies (often in triclinic) for different crystal systems; (5) the presence of some elements, for example, transition metals required on average high k -point densities; the presence of electronegative elements (O and F) required high cut-offs. The results imply that the derived convergence data would be expected to depend strongly on the pseudopotential used; therefore one cannot use them for calculations with Quantum ESPRESSO and Quantum ESPRESSO's set of pseudopotentials.

Janssen *et al.*¹⁰ conducted a systematic study aimed at understanding the optimal way to choose energy cutoffs and k -point density for DFT calculations with VASP (so smearing was fixed to default VASP values). They considered energy-volume dependence and derived quantities (the bulk modulus in particular as it is more sensitive to fitting/convergence errors) as the target properties to analyse convergence in several elemental metal compounds. They decomposed the total error into a systematic part, obtained by comparing results at finite cutoffs and k -point meshes against highly converged references, and a statistical part, identified as the residual oscillatory noise arising from discrete changes in the number of plane waves or k -points. This separation enabled the construction of error phase diagrams, distinguishing regimes where systematic or statistical errors dominate. Their analysis showed that systematic errors (*e.g.*, from insufficient cutoffs) dominate for most elements, while statistical errors can prevail in transition metals such as Cu, Pd, and Ag. They concluded that in the statistical regime, it is most efficient to refine the cheaper parameter (typically the k -points), whereas in the systematic regime, both the cutoff energy and k -point density must be increased simultaneously.



The Quantum ESPRESSO community paid a lot of attention to the systematic verification of pseudopotentials and pseudo-potential-based codes. The verified codes included those using different numerical basis sets, such as plane waves, Gaussian functions combined with plane waves, Daubechies wavelets, and atomic orbitals.^{11–23} They designed a testing protocol (SSSP – Standard Solid-State Pseudopotential protocol) to benchmark pseudopotentials in Quantum ESPRESSO across multiple tasks (equations of state, convergence of phonon frequencies, cohesive energies, pressures, and band structures with respect to plane-wave cutoffs). They tested available pseudopotentials on elemental crystals (with rare-earth nitrides and SiF₄ being the exceptions). These investigations resulted in the list of recommended pseudopotentials (including ultrasoft, norm-conserving, and PAW) for each element in the periodic table and a recommended energy/density cut-off pair for these pseudopotentials. The authors also pointed out that the required cut-off depends strongly on the property of interest. They also noted the independence of convergence patterns for local/semi-local functionals (whereas for hybrid/meta-GGA, no dedicated pseudopotentials existed).

Nascimento *et al.*²⁴ extended this line of work by analysing the interplay between *k*-point sampling and electronic smearing in Quantum ESPRESSO calculations. Building on the SSSP pseudopotential library—with its recommended plane-wave and charge-density cutoffs—they carried out a systematic study on a dataset of 269 experimentally known compounds (including oxides, metals, and insulators, both 3D and 2D) selected from the Materials Cloud databases, with unit cells containing up to 14 atoms. Their goal was to determine how different choices of *k*-point density and smearing affect both numerical accuracy and the fraction of calculations that fail to converge. The guiding principle was that sampling (statistical) errors from insufficient *k*-points should be suppressed in favour of controlled systematic errors introduced by finite smearing, since the latter are predictable and tunable. Using Marzari–Vanderbilt cold smearing, they identified three practical protocols—fast, balanced, and stringent—that offer trade-offs between efficiency and precision. The balanced protocol (smearing temperature $\sigma = 0.02$ Ry, *k*-point spacing $\lambda = 0.15 \text{ \AA}^{-1}$) was shown to work well for most systems, while the stringent protocol ($\sigma = 0.0125$ Ry, $\lambda = 0.1 \text{ \AA}^{-1}$) was necessary for challenging cases such as lanthanides and actinides. In this way, they provided ready-to-use parameter sets that minimize errors and computational cost while ensuring robust convergence across large, chemically diverse datasets.

In the past year, there has been rapid progress in LLM-based multi-agent systems designed for automated computational chemistry workflows, such as El Agente,⁵ DREAMS (Density Functional Theory Based Research Engine for Agentic Materials Simulation),⁶ and VASPilot.⁷ These agents typically employ multiple child agents to handle automated convergence schemes, either by leveraging existing recommendations from the literature and packages such as pymatgen or ASE or by generating a series of calculations with systematically varied parameters for convergence testing. They offer clear advantages in managing complex tasks—such as input file generation, job

submission, error handling, and post-processing—without human intervention. However, a persistent challenge remains in determining the optimal set of calculation parameters that ensures accurate results while minimizing computational cost.

In summary, previous studies have progressively clarified the roles of cutoffs, *k*-point density, and smearing in determining the accuracy and efficiency of plane-wave DFT calculations. Nevertheless, the optimal choice of these parameters for high-throughput studies remains an open problem. In this work, we continue our work in this direction.

2 Data generation and analysis

2.1 Data generation

With all interconnected factors affecting *k*-point convergence for single-point energy calculations in mind, the following set of parameters was chosen to generate the data:

(1) The valence electrons are represented using the Standard Solid-State Pseudopotential (SSSP) library (version 1.3, PBEsol, efficiency set), with plane-wave cutoff energies determined following the efficiency-level recommendations.

(2) The Marzari–Vanderbilt cold-smearing scheme, with a smearing width of 0.01 Ry, is applied in all calculations. The choice of the cold smearing is justified by its ability to yield a free energy that is quite insensitive to variations in the smearing temperature. The smearing temperature is set to 0.01 Ry, which ensures smearing convergence for most materials according to Nascimento's report.²⁴ This also implies that even for insulators, smearing is applied to the electronic states around the Fermi level, rather than using fixed occupations.

(3) The 20 187 structures were selected randomly from the MC3D PBEsol-v1 dataset, and no additional structural relaxation was performed.²⁵

(4) Non-spin-polarized calculations were performed for all materials, and no magnetic configurations were taken into consideration.

(5) The *k*-mesh convergence is determined as follows: when the energy difference among three consecutive *k*-meshes becomes smaller than 1 meV per atom (an illustrative example is provided in Section 3 of the SI), the first point of these three is identified as the converged *k*-mesh. In the present workflow, the convergence criterion was defined using total-energy differences only, which means that atomic forces were not included in the dataset. This procedure does not guarantee that one has the optimal set of parameters, that all pockets of the Fermi surface are resolved, or that the total energy of compounds with a band gap below 0.14 eV is resolved correctly. However, it is an easy way to generate the data.

(6) We employ the definition of the *k*-point distance, as implemented in the AiiDA–quantumESPRESSO package,^{1,26} to generate various *k*-point meshes. The *k*-distance represents the maximum spacing (in \AA^{-1}) between adjacent *k*-points in reciprocal space, such that the number of *k*-points along each reciprocal lattice vector b_i is given by $\lfloor |b_i|/k_{\text{dist}} \rfloor$. Starting from a *k*-distance of 1.0 \AA^{-1} , we systematically scan all possible *k*-meshes by varying the *k*-distance in steps of 0.005 \AA^{-1} (an illustrative example is provided in Section 3 of the SI). Throughout this



work, we adopt Γ -centred meshes as a practical and consistent workflow choice, mainly to maintain compatibility with common follow-up calculations such as electronic-structure and phonon workflows.

The generated data were processed to form the dataset. This included reducing all systems to primitive cells and then removing all structural duplicates. To do this, we used pymatgen.²⁷ The property that we predict (k -points distance) is intensive and should not depend on the size of the number of formula units or centering of the cell. Pre-processing ensures that the model predictions are consistent with this physical constraint. The number of k -points that enter the Quantum ESPRESSO input file is easily generated for any unit cell/supercell from the k -point distance.

2.2 Data analysis

The final dataset comprises 20 178 unique crystal structures, encompassing all non-radioactive and some radioactive elements (see element occurrence frequencies in the SI, SI Fig. S1). The dataset exhibits a broad distribution in the number of atoms per primitive unit cell (SI Fig. S2) and includes compounds containing up to seven distinct elements (SI Fig. S3). Among the 230 crystallographic space groups, 200 are represented. Since the shape of the first Brillouin zone is determined by the Bravais lattice type, it is noteworthy that all 14 Bravais lattices are represented in the dataset (Fig. 1), ensuring comprehensive coverage of symmetry types relevant for k -point sampling.

Fig. 2 shows the distribution of compounds with respect to the target property (converged k -point distance) used for training the machine-learning models. The vertical dashed line indicates the recommended reference value of 0.06 \AA^{-1} reported by Bosoni *et al.*,¹¹ which is consistent with our convergence dataset: in most cases, the explicitly determined converged k -point spacing is larger than 0.06 \AA^{-1} , indicating that this literature value represents a conservative lower bound for reliable reference calculations. The overall distribution exhibits two distinct peaks, corresponding to metallic systems (lower maximum k -point distance) and insulators (higher distance). Fig. 3 presents analogous distributions for subsets of compounds constrained to individual Bravais lattices. There is a clear trend that more symmetric lattices, on average, require

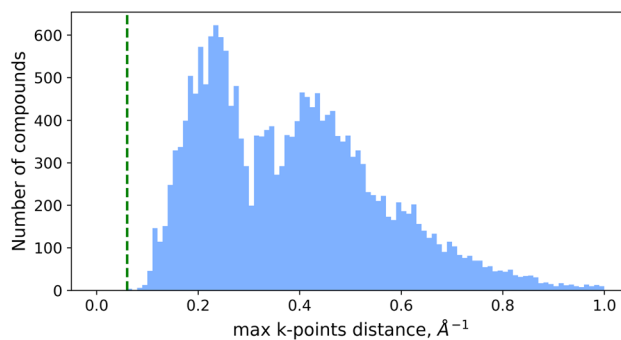


Fig. 2 Distribution of compounds with respect to the converged k -point distance in the dataset. The vertical dashed line represents the max k -point distance of 0.06 \AA^{-1} , recommended for the reference calculations reported by Bosoni *et al.*¹¹

dense k -point meshes. So, we can conclude that both metallicity and lattice symmetry significantly influence the optimal k -point density.

To illustrate the efficiency of using an optimized k -mesh relative to commonly used databases, we compare it with several widely adopted materials datasets. For example, the MC3D PBESol-v1 dataset²⁵ employs a default k -point distance of 0.15 \AA^{-1} , which is substantially denser (*i.e.*, over-converged) for approximately 93.4% of the materials in our generated database. The Materials Project²⁸ uses a default k -point density of $1000/(\text{number of atoms in the unit cell})$, which likewise results in a denser-than-necessary k -sampling for about 70% of our materials. Assuming identical settings for pseudopotentials, smearing, and symmetry reduction (as in MC3D PBESol-v1), an optimized, convergence-verified k -mesh would enable significant reductions in symmetrized k -point counts—and therefore in overall computational energy consumption, as shown in Fig. 4.

To illustrate how our optimized k -meshes compare with commonly used database defaults, we benchmark them against several widely adopted materials datasets. For example, the MC3D PBESol-v1 dataset²⁵ employs a default k -point distance of 0.15 \AA^{-1} , which yields denser k -sampling than our convergence-verified setting for approximately 93.4% of the materials in our generated database. The Materials Project²⁸ uses a default k -point density of $1000/(\text{number of atoms in the unit cell})$, which

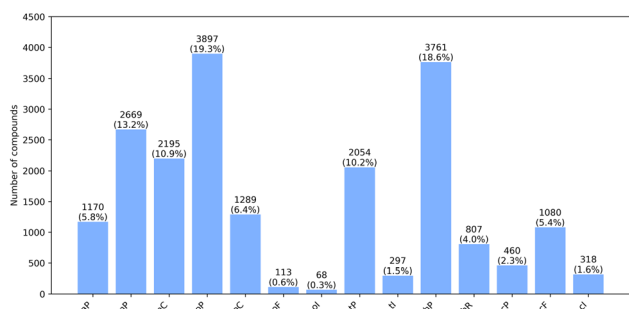


Fig. 1 Distribution of compounds in the dataset over 14 Bravais lattices.

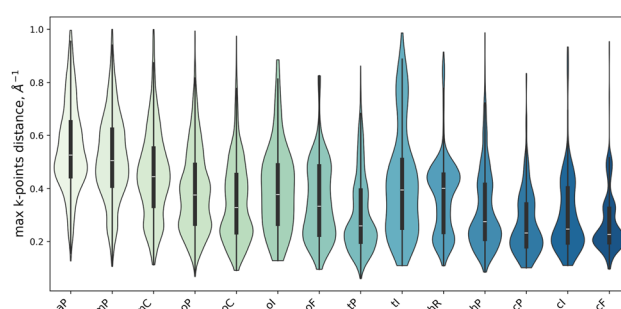


Fig. 3 Separate distributions with respect to the maximum distance between k -points for compounds with a fixed Bravais lattice.



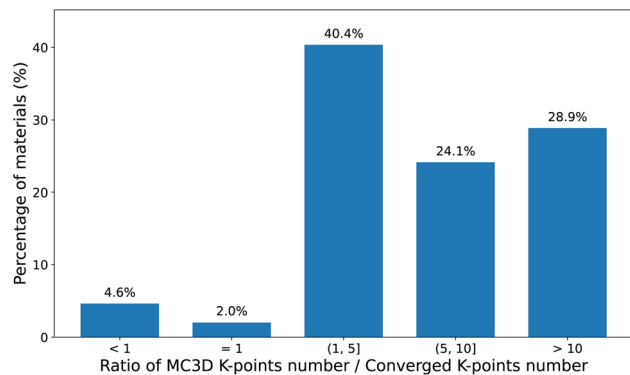


Fig. 4 Distribution of the ratio between the MC3D reduced k -point number (corresponding to a k -distance of 0.15) and the converged reduced k -point number across our dataset. Percentages above the bars indicate the fraction of materials in each interval.

likewise gives denser k -sampling than our converged setting for about 70% of the materials considered here. To better quantify the extent of this difference, Fig. 4 shows the distribution of the ratio between the default database k -point number and the converged k -point number. This indicates that the default settings are often more conservative than required by our convergence protocol and, in many cases, correspond to a substantially larger k -point-related workload. However, we note that the actual runtime impact depends on additional factors beyond the k -point number alone. In practice, the realized savings depend on how the electronic-structure code distributes work across parallelisation layers, including k -point pools, band groups, as well as on the communication characteristics of the underlying HPC architecture. Nevertheless, the reduction in required k -point sampling reported here remains a useful code-level proxy for potential efficiency gains across high-throughput workflows.

To evaluate the robustness of our non-spin-polarized calculations, we selected 268 materials evenly sampled across different categories, including metallic and semiconducting systems, the 14 Bravais lattice types, and four ranges of atomic counts (0–10, 10–20, 20–40, and >40 atoms per cell) for spin-polarized tests, for which the initial magnetic moments were set according to the MC3D protocols.²⁵ As presented in Fig. 5, the results show that 82% of these materials converge to the same k -mesh, while the remaining 18% exhibit magnetic behaviours in the spin-polarized calculations. As we are unable to capture the magnetic behaviours in non-spin-polarized calculations, it is understandable that the converged k -mesh is different. We also notice that among all samples, 94% have denser k -meshes in the non-spin-polarized calculations than in the spin-polarized calculations, while 6% have slightly lower densities, indicating that the non-spin-polarized calculations generally provide well-converged results. We also plan to extend the dataset with spin-polarized calculations and to develop a separate machine-learning workflow to identify magnetic materials and adapt the k -point recommendation accordingly.

As presented in Fig. 6, the SSSP Efficiency library (version 1.3) was employed in our training data generation, as it is

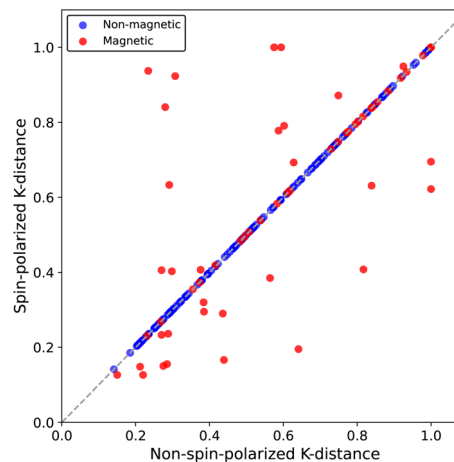


Fig. 5 Correlation between converged k -distances in non-spin-polarized and spin-polarized calculations.

recommended for high-throughput calculations with affordable computational cost.²⁹ We acknowledge that, in practical applications, users may prefer the SSSP Precision library to achieve results closer to all-electron calculations. This naturally raises the question of whether our dataset, trained with the Efficiency library, remains valid for predicting k -mesh settings when using the Precision library. To examine this, we selected 215 materials and performed non-spin-polarized calculations using the Precision pseudopotentials. The results show that in 90% of the cases, the Precision library yields a k -mesh that is either denser than or identical (78%) to that obtained with the Efficiency library. In the remaining 10%, 8% of the total materials show energy differences smaller than 2 meV/atom compared to the converged results, confirming the robustness of the Efficiency-trained dataset. More generally, transferability to other pseudopotential libraries or exchange–correlation treatments, including hybrid functionals, should not be assumed without further validation. To support this, we provide the full training and data-processing pipeline in our public repository, enabling users to retrain the models for alternative settings.

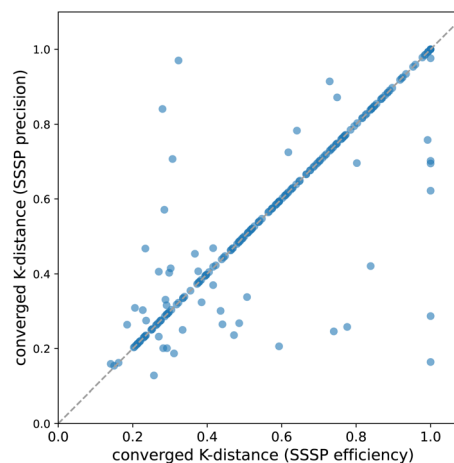


Fig. 6 Correlation between converged k -distances in SSSP-efficiency and SSSP-precision calculations.



In the discussions above, we identified the main limitation of our current non-spin-polarized database—its inability to correctly describe magnetic materials. However, determining whether a material is magnetic or non-magnetic is often non-trivial. To provide an initial assessment of this issue, we used the MC3D^{25,30} database as a reference to identify approximately 2800 potentially magnetic materials in our dataset and performed spin-polarized calculations on them with finite initial magnetic moments. The initial magnetic moments were assigned based on the presence of d or f electrons: materials containing such elements were initialized with their maximum possible magnetic moments (*i.e.*, 5 or 7 μB for d or f shells, respectively), while all others were assigned a small default value of 0.1 μB .²⁵ We found that only about 43% of the non-spin-polarized calculations yield the same converged k -mesh as their corresponding spin-polarized counterparts for these magnetic materials, indicating that for magnetic materials, spin polarization can significantly affect the convergence behaviour and, consequently, the accuracy of the derived properties.

3 Developing prediction models

3.1 Models to predict k -point density

To predict the density of k -points, we train several models on our dataset: Random Forest Regression, Gradient Boosting Regression, CrabNet, CGCNN, and ALIGNN.

Random Forest Regression³¹ (RF) is a machine learning method that uses an ensemble of multiple (*i.e.* 100) decision trees, each trained on a bootstrap sample of the data with a random subset of features. The final prediction is obtained by averaging the predictions of individual trees, which reduces variance and overfitting. We use the Random Forest Regressor implemented in the scikit-learn³² library.

Gradient Boosting Regression³³ (GBR) is another ensemble-based method, which, instead of averaging the predictions of multiple weak learners, builds them sequentially, where each subsequent tree attempts to correct the errors of the previous one. We use the Gradient Boosting Regressor implemented in scikit-learn.

The Random Forest Regressor and Gradient Boosting Regressor require features in a tabular format representing each compound as a vector derived from the compound's compositions or/and structure. We use the following sets of features:

(1) C = composition features. These are essentially the Magpie³⁴ style features computed with the matminer library.³⁵ We include ElementProperty, Stoichiometry, and ValenceOrbital. ElementProperty features are composed of statistics [“minimum”, “maximum”, “range”, “mean”, “avg_dev”, and “mode”] of the properties of atoms in the structure. The examples of properties are number, Mendeleev number, atomic weight, electronegativity, *etc.* Stoichiometric features depend only on the fractions of elements; these include the number of elements in the compound and several L^p norms of the fractions. ValenceOrbital features represent the average number and/or fraction of valence electrons in specified orbitals.

(2) S = structural features. For the first part of these features, we use GlobalSymmetryFeatures and DensityFeatures from

matminer. These include space group number, crystal system, centrosymmetry, number of symmetry operations, density, volume per atom, and packing fraction. For the second part of these feature vectors, we use modified Smooth Overlap of Atomic Positions (SOAP) fingerprints.³⁶ The modification consists of: (1) substituting all atoms with an ‘X’ atom type; (2) calculating SOAP fingerprints for the structure with anonymised atoms with the DScribe library;³⁷ (3) averaging the resulting SOAP descriptors over all atoms. This modification is required to ensure that the features have the same size for all compounds.

(3) L = lattice features. Lattice features include a , b , c , α , β , and γ parameters for the unit cell, the inverse unit cell, and the Bravais lattice.

(4) JarvisCFID features were originally suggested for the prediction of k -point length by Choudhary and Tavazza⁹ and are available in a matminer library.³⁵ These are complex feature vectors with a length of 1557, which include information about composition, unit cell, core charges, angular distribution function, radial distribution function, dihedral angle distribution function, and nearest neighbours.

(5) M = metallicity features. It is expected that the required k -point density strongly depends on whether a compound is metallic or insulating. In metals, a denser k -point mesh is required to accurately resolve the Fermi surface, whereas in semiconductors and insulators, the Brillouin zone integration converges more rapidly. The reason is that metallic systems have a discontinuity in electronic occupation at the Fermi level, making the total energy highly sensitive to how finely the electronic states are sampled. If the mesh is too coarse, errors in the total energy can reach tens of meV per atom, leading to unreliable stability comparisons and inaccuracies in derived properties such as formation energies or phonon spectra. Since metallicity is not known *a priori* for arbitrary compounds, we train a dedicated CGCNN model on Materials Project data to classify materials as metals or non-metals. By performing inference with a trained model for a new compound and extracting learned embeddings, we get numerical vectors which encode information about metallicity for each particular compound, and can be used as additional features in k -point prediction models. So, M features are those internal embeddings from the pre-trained CGCNN model. (The ‘is_metal’ dataset was downloaded from MP in July 2025 and contained ~180 000 unique structures. Performance of the trained CGCNN model on the test set is as follows: accuracy = 0.84, F1_score = 0.83, Matthews Correlation Coefficient = 0.69. The model checkpoint can be found in the code repository.)

CrabNet³⁸ is a composition-based deep learning model that adapts the Transformer architecture,³⁹ originally developed for natural language processing, to materials science applications. In CrabNet, the elements in a chemical formula serve as tokens, analogous to words in a sentence. Each element is associated with a learned embedding vector, while stoichiometric fractions are incorporated through a modified positional-encoding scheme that enables the model to represent not only which elements are present but also their relative proportions. Passing these token-fraction embeddings through stacked self-attention



layers allows CrabNet to capture complex inter-element interactions, modelling both pairwise and higher-order chemical relationships.

As input features, we use mat2vec embeddings⁴⁰ together with extended cgcnn descriptors, cgcnn + edc. Mat2vec embeddings were developed by Tshitoyan *et al.*⁴⁰ via a skip-gram variation of the Word2vec method trained on 3.3 million scientific abstracts and were originally used in the CrabNet model. CGCNN features were suggested in the CGCNN paper⁴¹ and include elemental attributes such as atomic number, atomic weight, and basic electronic properties. The extended cgcnn feature set (cgcnn + edc) augments the original cgcnn elemental descriptors with information from the SSSP 1.3 pseudopotential library (efficiency, PBEsol), including recommended energy and density cutoffs and the pseudopotential type (norm-conserving, ultrasoft, *etc.*). These quantities are element-specific because they depend on the corresponding pseudopotential. Their inclusion is motivated not by a fundamental dependence of k -point convergence on cutoff values, but by the practical observation that, in finite-accuracy high-throughput calculations, cutoff-related quantities can provide useful complementary information.⁹ In a style similar to the cgcnn features, we encode the numbers for cut-off values as one-hot feature vectors of a size suitable for the range of features. Finally, to reduce redundancy in cgcnn + edc descriptors, we perform PCA on them and remove dimensions with negligible information content.

The next model we try is the CGCNN,⁴¹ or the Crystal Graph Convolutional Neural Network, which is a graph neural network designed to represent crystalline materials as atom graphs. Here, we consider two ways to build an atom graph: (1) a radius graph, which assumes that two atoms are connected by an edge if the distance between them (or their periodic images) is smaller than a cutoff radius. Such a graph usually is a multi-graph; (2) a Voronoi tessellation graph, which assumes that two atoms are connected by an edge if they share the face of the Voronoi polyhedra. Once the atom graph is constructed, the CGCNN employs a gated graph convolution operation for message passing, allowing atom embeddings to be updated based on their local chemical environments. By stacking multiple convolutional layers, the model captures both short- and longer-range structural information, making it effective for predicting a wide variety of material properties directly from crystal structures.

The ALIGNN⁴² conceptually follows the CGCNN model and extends this framework by introducing a line graph in addition to the atom graph, thereby explicitly encoding angular information between neighbouring bonds. This enriched representation enables the model to distinguish between the graphs that would be indistinguishable using only information about the atoms' identities and distances between them.

As input features for structure-based GNN models, we employ cgcnn features, extended cgcnn features (cgcnn + edc), and mat2vec features. In the study⁴³ it was shown that, due to the limited power of the considered GNNs in representation, the GNN models fail to learn some properties of materials, which are commonly used as physical descriptors. The authors

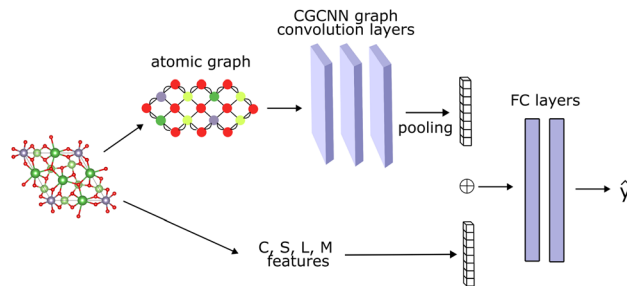


Fig. 7 The schematic of the CGCNNd model.

performed a thorough analysis of the issue and advocated for injecting the descriptors that are difficult for GNN models to learn after graph convolution layers. These include mainly features related to periodicity and symmetry, such as unit cell parameters, space group, crystal system, *etc.* As these properties are important for the prediction of the k -point distance, we use the modification suggested in (ref. 43) for our GNN models. The resulting models are named CGCNNd and ALIGNNd, respectively. The schematic of the CGCNNd model is shown in Fig. 7. To inform GNN models about the metallic properties of the compounds, we also use our metallicity features in CGCNNd and ALIGNNd.

3.2 Estimation of uncertainty

Predicting the k -point distance with machine-learning models inevitably introduces errors. These errors affect DFT calculations asymmetrically. Underestimation of the k -point distance and ensuing oversampling of the Brillouin zone, is safe, since the SCF calculation still lies in the converged region. Conversely, overestimation of k -distances would lead to under-sampling of the Brillouin zone and unreliable results. To reduce the probability of overprediction, we shift the prediction target from the median (50th percentile) to the lower conditional quantile of the k -point distance distribution.

Prediction uncertainty has two main sources: (1) aleatoric uncertainty, which represents statistical noise in the data; (2) epistemic uncertainty, which arises due to uncertainty in the model with which we fit the data; this uncertainty arises due to the finite number of points in the training or model misspecification. Since our objective is to estimate lower quantiles of the required k -point distance to avoid overestimation, we primarily focus on aleatoric uncertainty.

To estimate aleatoric uncertainty, we employ conformalised quantile regression,⁴⁴ which we found to be the most effective approach for our application. This method inherits the strongest features of the predecessor methods: quantile regression and conformal prediction.

Quantile regression learns the conditional quantiles of a dependent variable as a linear function of the explanatory variables $Q_\alpha(x) = y: F(y|x) \geq \alpha$. To perform quantile regression, one uses a special asymmetric, pinball loss function:

$$L_\alpha(y_i, \hat{y}_i(x_i)) = \begin{cases} \alpha(y_i - \hat{y}_i(x_i)), & \text{if } y_i > \hat{y}_i(x_i), \\ (1 - \alpha)(\hat{y}_i(x_i) - y_i), & \text{if } y_i \leq \hat{y}_i(x_i). \end{cases} \quad (1)$$



This way, one α -quantile is learned. We use this loss to train the ALIGNN and CGCNN models to predict conditional quantiles.

For a Random Forest model, quantiles are estimated in another way.⁴⁵ Conditional quantiles can be inferred with Quantile Regression Forests (QRF). QRF are usually trained with a standard MSE loss and is a generalisation of random forests, which retain for each leaf, the empirical distribution of training targets rather than only their mean. The empirical distribution is the weighted distribution of observed response variables, where the weights attached to observations are identical to those the original random forest algorithm. And from the estimation of conditional distribution $\hat{F}(y|x = X)$, the α -quantile $Q_\alpha(x)$ is obtained as follows:

$$Q_\alpha(x) = \inf\{y:F(y|X = x) \geq \alpha\} \quad (2)$$

We use the implementation of the Quantile Regression Forests from the sklearn-quantile package.

Conformal prediction (CP) is a distribution-free framework that provides valid prediction intervals with valid finite-sample marginal coverage under the assumption of exchangeability of the samples. Unlike parametric methods, it does not rely on strong distributional assumptions and can be applied as a wrapper around any machine learning method. Given a regression model $f(x)$, CP constructs a prediction interval:

$$C(x_{n+1}) = [f(x_{n+1}) - \beta_\alpha, f(x_{n+1}) + \beta_\alpha] \quad (3)$$

where:

$$\beta_\alpha = Q_{(1-\alpha)(1+1/|C|)}(\{\beta_i\} \text{ for } i \in C) \quad (4)$$

is an empirical quantile of residuals $\beta_i = |y_i - f(x_i)|$ on the calibration set C .

Conformalised Quantile Regression (CQR) preserves the strong features of quantile regression and conformal predictions and provides theoretical guarantees of distribution-free finite-sample coverage with asymmetric, heteroscedastic intervals conditioned on data points. CQR begins with a model that predicts the lower and upper conditional quantiles $Q_{\alpha_{lo}}(x)$ and $Q_{\alpha_{hi}}(x)$. Then, conformity scores are calculated on calibration dataset C : $\beta_i = \max\{Q_{\alpha_{lo}}(X_i) - Y_i, Y_i - Q_{\alpha_{hi}}(X_i)\}$. Then the empirical quantile β_α is calculated for the distribution of conformity scores as in the CP method. And the final confidence intervals are determined as follows:

$$C(X_{n+1}) = [Q_{\alpha_{lo}}(X_{n+1}) - \beta_\alpha, Q_{\alpha_{hi}}(X_{n+1}) + \beta_\alpha] \quad (5)$$

4 Results

4.1 Overall comparison of performance of the models

First, we train the models in combination with different sets of features and different ways to construct atomic graphs to predict the mean value of the k -point distance. For all models the data were split into training, validation, and testing datasets in an 80 : 10 : 10 proportion. We perform a random search over

the hyperparameter space on the validation dataset for all models. Then, for the best set of hyperparameters, we perform a manual hyperparameter search in the vicinity of it. For the final hyperparameter sets, all models are trained on 3–5 different data splits determined for randomly chosen random seeds, and their performance is averaged. Confidence intervals for performance are estimated under the assumption of a Gaussian distribution of errors in the performance metrics.

We evaluate model performance using mean absolute error (MAE), mean absolute percentage error (MAPE), mean squared error (MSE), the coefficient of determination (R^2), the Spearman rank correlation coefficient, and the Kendall rank correlation coefficient. The inclusion of rank-based metrics (Spearman and Kendall) is motivated by the need to assess how well the models preserve the correct pairwise ordering of k -point densities. This allows us to quantify whether the ML models introduce additional randomness into the relative ordering of the samples, even when absolute errors remain small.

Table 1 shows the performance of the ensemble models, RF and GB, trained with different feature sets. In the table, the feature sets are C = composition features (size = 146), CS = composition + structure + SOAP (size = 404), CSL = composition + structure + SOAP + lattice (size = 419), CSLM = composition + structure + SOAP + lattice + metallicity (size = 483), and JarvisCFID (size = 1557). One can see that the performance of Random Forest in general is better than that of GB on this task. The best model, Random Forest + CSLM, achieves an R^2 score of 0.703, MAE = 0.067, MAPE = 0.197, Spearman correlation = 0.86. Fig. 8 shows the scatter plot for predictions made by this model.

Comparing the performance of the ensemble models with different feature sets, we can see that each subset of features contributes to the final result. To quantify the contribution of each block of features to the performance of the final model, we make predictions on modified representations of the samples from the test set. We shuffle features in each block of features and see how much the performance of the prediction drops. The results are shown in Table 2. It can be seen that each set of features contributes to the final prediction, with composition features having the largest contribution, followed by metallicity features and lattice features. From the viewpoint of general physics/chemistry knowledge, this rating seems to be reasonable and supports the considerations we used when constructing the list of features.

Next, we turn to the training of the CrabNet model, which is based on transformer architecture and takes composition information as the only input. CrabNet was trained with mat2vec and extended cgcnn features (cgcnn + edc). Notably, the performance of this composition-only model is significantly worse than that for models with some encoding of structural information. For k -point prediction models, one would expect this kind of behaviour for models that capture the essence of the task (Table 3).

Table 4 shows the performance of the GNN models. The CGCNN model was trained with cgcnn features and two ways to construct an atomic graph, standard radius graph (rg + cgcnn) and crystalline graph, which has edges only between atoms



Table 1 Performance of different ensemble models

#	Model	Features	MAE	MAPE	MSE	R^2	Spearman	Kendall
1	RF	C	0.075 ± 0.0004	0.221 ± 0.003	0.0109 ± 0.0002	0.641 ± 0.005	0.821 ± 0.004	0.629 ± 0.002
2	RF	CS	0.0695 ± 0.0006	0.206 ± 0.005	0.0095 ± 0.0001	0.686 ± 0.008	0.850 ± 0.004	0.661 ± 0.005
3	RF	CSL	0.067 ± 0.002	0.197 ± 0.006	0.0091 ± 0.0003	0.702 ± 0.009	0.861 ± 0.006	0.674 ± 0.008
4	RF	CSLM	0.067 ± 0.001	0.197 ± 0.006	0.0090 ± 0.0003	0.703 ± 0.008	0.861 ± 0.006	0.676 ± 0.006
5	RF	JarvisCFID	0.078 ± 0.0007	0.234 ± 0.005	0.0113 ± 0.0002	0.628 ± 0.003	0.815 ± 0.004	0.618 ± 0.004
6	GB	C	0.080 ± 0.002	0.240 ± 0.005	0.0118 ± 0.0004	0.616 ± 0.007	0.804 ± 0.004	0.605 ± 0.004
7	GB	CS	0.0741 ± 0.0008	0.219 ± 0.004	0.0102 ± 0.0003	0.659 ± 0.009	0.833 ± 0.005	0.636 ± 0.005
8	GB	CSL	0.0731 ± 0.0008	0.214 ± 0.002	0.0100 ± 0.0003	0.668 ± 0.008	0.839 ± 0.005	0.642 ± 0.006
9	GB	CSLM	0.074 ± 0.002	0.218 ± 0.005	0.0102 ± 0.0005	0.660 ± 0.010	0.833 ± 0.008	0.640 ± 0.008
10	GB	JarvisCFID	0.0803 ± 0.0008	0.236 ± 0.005	0.0119 ± 0.0004	0.610 ± 0.010	0.807 ± 0.008	0.606 ± 0.006

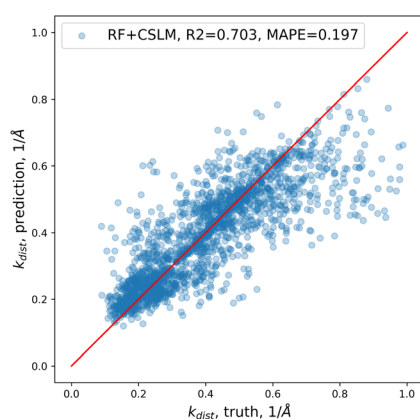


Fig. 8 Truth vs. prediction for the best ensemble model RF + CSLM on the test set.

Table 2 Performance of RF + CSLM on the test set described by modified features. Each feature modification represents feature shuffling in one of the feature blocks

#	Shuffled block	MAE	MAPE	MSE	R^2	Δ MAE	Δ R^2
1	—	0.067	0.197	0.009	0.703	0	0
2	Composition	0.102	0.347	0.017	0.455	-0.035	-0.248
3	Structure	0.068	0.192	0.009	0.693	-0.001	-0.01
4	SOAP	0.074	0.211	0.011	0.644	-0.007	-0.059
5	Lattice	0.074	0.204	0.011	0.626	-0.007	-0.077
6	Metallicity	0.078	0.248	0.011	0.612	-0.011	-0.091

sharing Voronoi surfaces (crg + cgcn). The standard radius graph performs better. Because of this we used the standard radius graph in all other cases. Next, we try different atomic input features for the CGCNN, the list includes cgcn (rg + cgcn), mat2vec (rg + mat2vec), and cgcn + edc (rg + cgcn + edc). From all these combinations the standard cgcn features

Table 3 Performance of CrabNet with different features

#	Model	Features	MAE	MAPE	MSE	R^2	Spearman	Kendall
1	CrabNet	mat2vec	0.084 ± 0.001	0.248 ± 0.007	0.0134 ± 0.0003	0.56 ± 0.01	0.772 ± 0.005	0.575 ± 0.005
2	CrabNet	cgcn + edc	0.084 ± 0.002	0.245 ± 0.004	0.0138 ± 0.0006	0.55 ± 0.02	0.77 ± 0.01	0.58 ± 0.01

give the best performance. So, for this combination we try to inject additional features after the GNN layers. The schematic of the extended CGCNN model is shown in Fig. 7. The performance of the extended CGCNN model is slightly improved compared to the vanilla model, both in the case of CSL combination of features (CSL = composition + structure (without SOAP, just matminer) + lattice) and metallicity features (m). The ALIGNN model was trained with the radius graph only (for construction of both atomic and linear graphs) with cgcn and cgcn + edc features, with the latter showing slightly better performance. So, we choose this combination for the extended ALIGNN model taking in additionally CSL or metallicity features. The performance of the ALIGNN + cgcn + edc + CSL model is comparable to that of our best Random Forest model (Fig. 9).

Across all models tested, the ensemble methods (Random Forest and Gradient Boosting) outperform composition-only neural models (CrabNet) while performing comparably to advanced graph neural networks (ALIGNN). The best-performing model overall is RF + CSLM, which reaches an R^2 of 0.7, slightly surpassing the ALIGNN + cgcn + edc + CSL model.

4.2 Interpretation of k -point prediction results

To analyse the factors affecting the decision about k -point distance value, we train a surrogate model,⁴⁶ which is a shallow decision tree based on the predictions of the RF + CSLM model. As feature vectors are very high-dimensional, we do this in two stages. First, we train RF + CSLM on different data splits and, for each of them, fit a surrogate model. These decision trees have a lot in common, but also containsome features that are different between different data splits. We collect all features that appear in those surrogate decision trees and reduce the features to only those that appear in the surrogate model in composition, structure, and lattice block. We then perform PCA



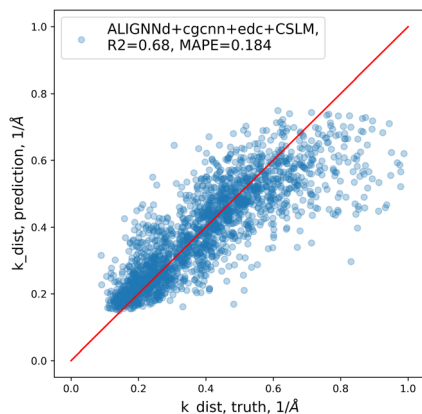


Fig. 9 Truth vs. prediction for ALIGNNd + cgcn + edc + CSLM on the test set.

reduction to 5 features for the SOAP and CGCNN blocks. The new features have a size of 20 instead of 483. Next, we refit the surrogate decision trees on these reduced feature sets to obtain simple sets of decision rules for k -point distance prediction. The R^2 -score for the final surrogate model with a depth of 4 and minimum 40 samples per leaf is 0.706 (we compare the

predictions of the surrogate model with predictions of the fitted RF + CSLM model here, not the true values).

An example of the final decision tree is shown in Fig. 10. One can see that the root is melting temperature. The melting temperature is not a true melting temperature, but a feature that is calculated as a weighted mean of the melting temperatures of atomic compounds composed of atoms in the composition formula, and the weights are calculated from atomic fractions in the composition. And the splitting value is 203 K. It is an extremely low melting temperature; the elements with melting temperatures below this threshold are H, N, O, F, Cl, Ne, Ar, Kr, Xe, and Rn. So, the compounds with melting temperature below 203 are those rich in these elements, and surprisingly, approximately 50% of such compounds in the dataset fall in this category. Splitting at the melting temperature is a very stable feature appearing in all decision trees that we built for different splits of data.

Decision subtrees for compounds with melting temperatures below 203 K, and above this value, are very different. Following the 'yes' branch, we arrive at the condition 'space group number <38'. This is also a very stable feature that persists for all data splits. For a low-symmetry compound, the next feature to split upon is the mean atomic number. For high symmetry compounds, the split on the third level of the tree is on the first

Table 4 Performance of the GNN models

#	Model	Features	MAE	MAPE	MSE	R^2	Spearman	Kendall
1	CGCNN	rg + cgenn	0.074 ± 0.002	0.207 ± 0.004	0.0112 ± 0.0006	0.630 ± 0.010	0.827 ± 0.006	0.634 ± 0.008
2	CGCNN	crg + cgenn	0.077 ± 0.001	0.223 ± 0.006	0.0115 ± 0.0003	0.630 ± 0.010	0.819 ± 0.006	0.626 ± 0.005
3	CGCNN	rg + mat2vec	0.0748 ± 0.0008	0.208 ± 0.005	0.0116 ± 0.0002	0.620 ± 0.005	0.822 ± 0.003	0.629 ± 0.003
4	CGCNN	rg + cgenn + edc	0.075 ± 0.002	0.215 ± 0.004	0.012 ± 0.001	0.630 ± 0.020	0.819 ± 0.007	0.629 ± 0.008
5	ALIGNNN	rg + cgenn	0.0700 ± 0.0005	0.200 ± 0.001	0.0105 ± 0.0002	0.644 ± 0.008	0.837 ± 0.003	0.650 ± 0.003
6	ALIGNNN	rg + cgenn + edc	0.0717 ± 0.0007	0.194 ± 0.003	0.0106 ± 0.0002	0.654 ± 0.008	0.845 ± 0.004	0.656 ± 0.004
7	CGCNNd	rg + cgenn + CSL	0.0737 ± 0.0007	0.206 ± 0.005	0.0111 ± 0.0004	0.647 ± 0.005	0.841 ± 0.001	0.649 ± 0.001
8	CGCNNd	rg + cgenn + M	0.0740 ± 0.0002	0.209 ± 0.001	0.0117 ± 0.0008	0.630 ± 0.020	0.839 ± 0.004	0.648 ± 0.003
9	ALIGNNd	rg + cgenn + edc + CSL	0.069 ± 0.002	0.186 ± 0.004	0.0099 ± 0.0003	0.680 ± 0.010	0.856 ± 0.005	0.669 ± 0.008
10	ALIGNNd	rg + cgenn + edc + M	0.070 ± 0.001	0.190 ± 0.004	0.0104 ± 0.0002	0.662 ± 0.008	0.849 ± 0.003	0.661 ± 0.004

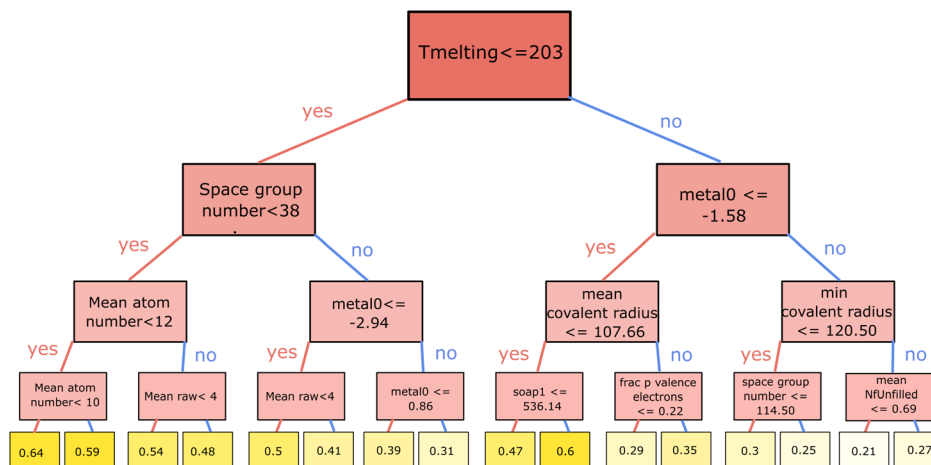


Fig. 10 Surrogate model for RF-CSLM predictions. Yellow rectangles are leaves with final values of the k -point distance.



PCA component (most informative) of the metallicity vector. The Metal0 feature carries information about the metallicity of the compound. However, the correspondence of the metallicity property of the compound is not fully described by this feature alone, and the meaning of the numerical value can't be interpreted easily. The final split in the decision tree is based on mean atomic number, mean atomic row, and metal0.

Following the 'no' branch, we arrive and the condition $\text{metal0} \leq -1.58$. The next level of the decision tree uses "mean covalent radius" or "min covalent radius" depending on the metallicity property, showing the importance of the covalent radius in decision making about the k -point distance. The final level of decision-making is composed of structural and electronic features.

Analysis of decision making by the RF-CSLM model shows that the k -point distance is determined by diverse behaviour, and apart from composition features, global symmetry descriptors, metallicity descriptors, and electronic configuration are important too. This conclusion additionally explains why CrabNet is so inefficient for this task, and why the GNN models perform worse than RF + CSLM.

4.3 Uncertainty estimation results

After identifying the best models, we perform uncertainty estimation for these models with conformalised quantile regression. Performing uncertainty estimation allows us to shift the prediction from the mean value to the α -quantile, ensuring that at least $\alpha \times 100\%$ of cases of predicted k -point distances correspond to dense enough meshes to ensure convergence. To evaluate the models making shifted predictions, we establish a baseline as a constant value of the k -point distance, chosen in a way that it is the largest possible k -point distance for which $\alpha \times 100\%$ of compounds are converged. This value is expected to be different depending on the chosen confidence level. We would like to stress that usually the training set is not available or very small, so the constant value of k -points for high-throughput calculations is chosen in a less rigorous way. So, the suggested baseline is strong. The mean error, showing the deviation of the used value from the true one, represents the measure of the accuracy of the models and translates into computational time in practical settings. Smaller errors correspond to more accurate k -point distance prediction and more computation time saved.

The results of CQR for ALIGNNd + cgcn + edc + CSLM are shown in Fig. 11–13 for CQR coverages of 0.9, 0.8, and 0.7. Light blue points and confidence intervals show the samples for which the true value is inside the confidence interval; dark blue points and confidence intervals show the samples for which the true value lies outside of the confidence interval. It can be seen that the CQR empirical coverage corresponds well to the target coverage. The mean size of the confidence intervals decreases with a reduction in coverage, as expected.

Table 5 shows the results of conformalised quantile regression for the best models. In all cases, the empirical coverage is consistent with the target coverage, and the confidence intervals have a reasonable size. This means that our models guarantee

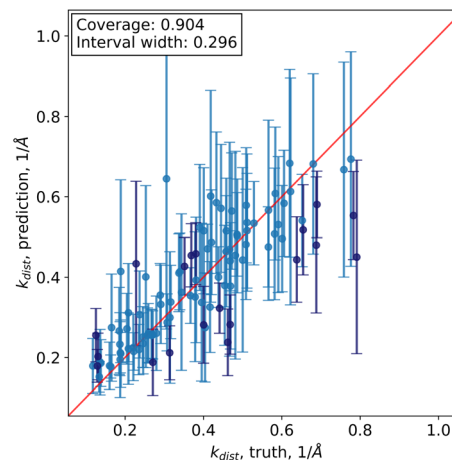


Fig. 11 CQR for ALIGNNd + cgcn + edc + CSLM, target coverage 0.9.

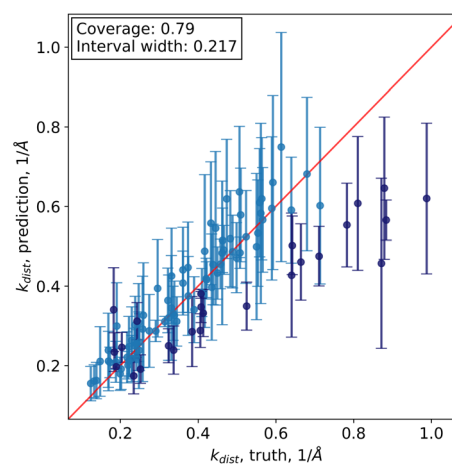


Fig. 12 CQR for ALIGNNd + cgcn + edc + CSLM, target coverage 0.8.

that a target fraction of compounds is calculated accurately, conditioned on training set distribution. ALIGNNd + cgcn + edc + CSLM provides slightly better coverage, narrower

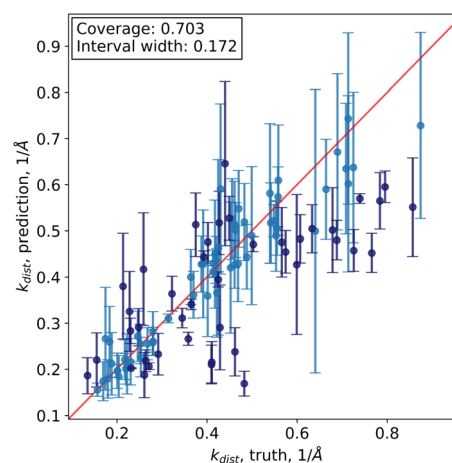


Fig. 13 CQR for ALIGNNd + cgcn + edc + CSLM, target coverage 0.7.



Table 5 Performance of the CQR models using different target coverages and target quantiles. Empirical coverage and MAE values are computed on the test set

#	Model	Features	Target coverage	Target quantile	CQR interval coverage	CQR width	Empirical coverage	MAE	k -distance baseline	MAE baseline
1	RF	CSLM	0.95	0.90	0.895	0.313	0.958	0.152	0.158	0.237
2	RF	CSLM	0.90	0.80	0.820	0.268	0.917	0.122	0.184	0.215
3	RF	CSLM	0.85	0.70	0.720	0.180	0.849	0.104	0.205	0.200
4	ALIGNNd	rg + cgcnn + edc + CSL	0.95	0.90	0.904	0.254	0.966	0.143	0.158	0.238
5	ALIGNNd	rg + cgcnn + edc + CSL	0.90	0.80	0.790	0.217	0.908	0.114	0.184	0.215
6	ALIGNNd	rg + cgcnn + edc + CSL	0.85	0.70	0.703	0.172	0.840	0.095	0.205	0.200

confidence intervals, and correspondingly smaller errors than RF + CSLM. All ML models at all confidence levels provide smaller overall errors than the baseline model. So, using ML models for the prediction of k -point distances is practically beneficial.

Fig. 14 and 15 show the distribution of errors for RF + CSLM (5th percentile), ALIGNN + cgcnn + edc + CSLM (10th percentile), and the baseline model for different quantiles. Dark blue colour shows the samples for which the predicted k -distance is larger than the true one. These figures show that distribution of errors for ML models is narrower than that for the baseline model, which results in smaller mean absolute errors.

Translation of the gain in k -distance prediction error into gains in computation time in general will depend on a lot of details, such as parallelisation strategy, hardware setup, materials calculated, *etc.* However, to demonstrate that the gain is present, we calculate the gain in terms of wall time using data that we obtained in the process of generating our dataset. The results are shown in Table 6. For our dataset, the mean wall time per sample is 352 s, which is calculated using the wall time corresponding to true converged k -distances. Both ML models adjusted to predict a quantile suitable for the required confidence, and the baseline model uses different k -distances, resulting in spending additional time on performing

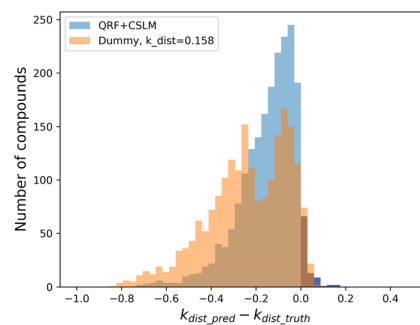


Fig. 14 Distribution of errors for the shifted RF + CSLM model and the baseline model. Target coverage is 95% of k -point distances. Dark blue bins show the samples for which the predicted k -distance is larger than the true one.

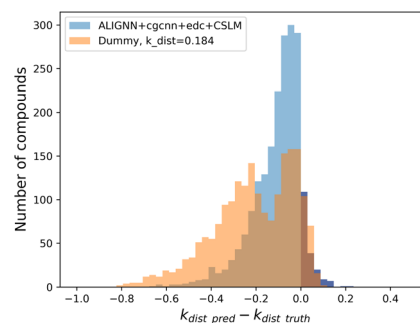


Fig. 15 Distribution of errors for the shifted ALIGNN + cgcnn + edc + CSLM model and the baseline model. Target coverage is 90% of k -point distances. Dark blue bins show the samples for which the predicted k -distance is larger than the true one.



Table 6 Wall-time analysis for predicted k_{dist} compared with optimal and dummy baselines. Excessive wall time is defined relative to the wall time at optimal k_{dist}

#	Model	Features	Mean wall time (optimal), s	Confidence level	Excessive wall time (pred), s	Excessive wall time (dummy), s	Gain per sample, s	Gain, %
1	RF	CSLM	352	0.95	388	497	109	22
2	RF	CSLM	352	0.90	277	318	41	13
3	RF	CSLM	352	0.85	240	223	17	7
4	ALIGNNd	rg + cgcnn + edc + CSL	352	0.95	368	468	100	21
5	ALIGNNd	rg + cgcnn + edc + CSL	352	0.90	247	311	64	21
6	ALIGNNd	rg + cgcnn + edc + CSL	352	0.85	217	217	0	0

calculations. We calculate the excessive wall time as the difference between the wall time corresponding to k -distance used by ML-model/baseline model and the mean optimal wall time in cases when the predicted k -distance is smaller than or equal to the true one (converged calculation), and in the opposite case all computation time is wasted, so all the time predicted by the model is added to the excessive time:

$$T_{\text{exc}} = \begin{cases} T_{\text{model}} - T_{\text{true}}, & k_{\text{dist}}^{\text{pred}} \leq k_{\text{dist}}^{\text{true}}, \\ T_{\text{model}}, & k_{\text{dist}}^{\text{pred}} > k_{\text{dist}}^{\text{true}}. \end{cases} \quad (6)$$

Interestingly, the gain in computation time for ML prediction compared to the baseline model depends on the quantile. For higher quantiles and more accurate calculations, the gain is more noticeable. With a decrease in the quantile, the number of the mispredicted samples grows, and the gain in time for ML predictions compared to the baseline model disappears.

5 Application for input file generation

To share the models we optimized, we developed a web application that allows generation of an input file for Quantum ESPRESSO single-point SCF total energy calculations, <https://goldilocks.streamlit.app/>.

The input page of the application asks the user to choose the variant of SSSP1.3 PBEsol to use, efficiency or precision, the ML model, and the confidence level. Our models are trained on data generated with SSSP1.3 PBEsol efficiency, but as shown above, for a large fraction of compounds, the k -point convergence values remain the same for SSSP1.3 PBEsol precision.

Next, the user is prompted to upload the structure CIF file or query the structure from one of the databases. Then, the structure may be subjected to transformations: reduced to a primitive one, or multiplied to create a supercell. When all parameters are specified, the user is prompted to generate the input file. Input file generation can be made with the ASE deterministic function or the LLM (the choice is available). The file generator makes prediction of the k -point distance using a primitive cell of the compound (as all structures in the training dataset were pre-processed into primitive ones), makes prediction of the k -point density, and then calculates the k -mesh from the inverse unit cell and the predicted k -point distance.

6 Discussion

The results presented above show that machine-learning models, when trained on a sufficiently large and diverse dataset of converged calculations, can reliably predict k -point densities for plane-wave DFT simulations. More importantly, they can do so with statistical guarantees when combined with conformalised quantile regression. This is particularly relevant for high-throughput and agentic workflows, where parameter selection is a major bottleneck.

However, several limitations remain. First, magnetic materials remain challenging: spin-polarized calculations often yield different k -mesh requirements, and our current dataset includes only non-spin-polarized data. Second, the dataset is tied to SSSP-1.3 PBEsol pseudopotentials; although we show that trends transfer to the Precision library, other pseudopotential families may require retraining. Third, while our models capture structural and compositional factors well, more expressive representations—such as equivariant GNNs or learned Brillouin-zone features—may further improve accuracy. Next, we did not try to pre-train GNN/CrabNet models on the metallicity dataset. Transfer learning could work better than feature injection. Finally, we could use a completely different approach for building an ML model: we could predict the total energy conditioned on the structure and other DFT parameters. Then the speed up of solving the convergence problem would be realized through using an ML surrogate to generate the convergence curve instead of predicting the converged k -distance. If successful, this approach could be applicable more generally. However, it is not immediately clear how to determine convergence in this case, as ML predictions of the total energy are expected to introduce errors, which could obscure the convergence point. Maybe a combination of both approaches could be used.

Despite these limitations, the current approach is successful in reaching the objective of improving over the constant baseline, which is currently a common choice. The integration with MCP servers also opens a path toward fully autonomous computational agents capable of end-to-end materials simulation.

7 Conclusions

In summary, we have developed a machine-learning framework that predicts the k -point density required for converged



Quantum ESPRESSO SCF energy calculations. Using a dataset of more than 20 000 convergence-tested materials, we benchmarked a broad set of models and demonstrated that both ensemble methods and graph neural networks achieve high predictive accuracy. Combining these models with conformalised quantile regression enables statistically guaranteed, underestimation-averse predictions.

Relative to fixed-density workflows, the model reduces unnecessary k -mesh sampling and thereby lowers computational cost while maintaining accuracy. The provided web platform makes this tool available. We also plan to release an MCP server in the near future to make our tool available in any MCP-compatible system.

Overall, this work establishes a practical and scalable route to data-driven parameter optimisation in DFT, opening doors to more efficient, greener, and more robust high-throughput materials simulations.

Author contributions

Elena Patyukova: conceptualization (AI), methodology (AI), data curation, investigation, formal analysis, software, validation, writing – original draft, visualization. Junwen Yin: conceptualization (DFT), methodology (DFT), data curation, investigation, writing – original draft, visualization. Susmita Basak: investigation, funding acquisition. Samuel Pinilla: investigation. Alin M. Elena: resources, funding acquisition. Gilberto Teobaldi: writing – review & editing, project administration, supervision.

Conflicts of interest

There are no conflicts to declare.

Data availability

The dataset generated and analysed in this study is available under a CC-BY-4.0 licence from UKRI-STFC eData at <https://doi.org/10.5286/edata/971>.

The archived version of the web application and associated source code used in this work is available from Zenodo at <https://doi.org/10.5281/zenodo.20324660>.

The archived version of the code used for training and evaluating the machine-learning models is available from Zenodo at <https://doi.org/10.5281/zenodo.20324627>.

The deployed web application is available at <https://goldilocks.streamlit.app/>.

Supplementary information (SI): additional details on the Goldilocks dataset, the QE input-generation application, k -distance and k -mesh generation, the convergence-definition example, the comparison with the JARVIS dataset, the $1/k$ -distance model test, and the metal/insulator k -distance distributions. See DOI: <https://doi.org/10.1039/d5dd00565e>.

Acknowledgements

This work was supported by EPSRC funding awarded to Barbara Montanari, Alin Elena, and Susmita Basak: Goldilocks

convergence tools and best practices for numerical approximations in Density Functional Theory calculations (EP/Z530657/1). This work was co-funded by the Ada Lovelace Centre at STFC-UKRI. Computing resources were provided by STFC Scientific Computing Department's SCARF cluster and STFC cloud.

Notes and references

- 1 S. P. Huber, S. Zoupanos, M. Uhrin, L. Talirz, L. Kahle, R. Häuselmann, D. Gresch, T. Müller, A. V. Yakutovich, C. W. Andersen, F. F. Ramirez, C. S. Adorf, F. Gargiulo, S. Kumbhar, E. Passaro, C. Johnston, A. Merkys, A. Cepellotti, N. Mounet, N. Marzari, B. Kozinsky and G. Pizzi, *Sci. Data*, 2020, 7, 300.
- 2 A. M. Ganose, H. Sahasrabudde, M. Asta, K. Beck, T. Biswas, A. Bonkowski, J. Bustamante, X. Chen, Y. Chiang, D. C. Chrzan, J. Clary, O. A. Cohen, C. Ertural, M. C. Gallant, J. George, S. Gerits, R. E. A. Goodall, R. D. Guha, G. Hautier, M. Horton, T. J. Inizan, A. D. Kaplan, R. S. Kingsbury, M. C. Kuner, B. Li, X. Linn, M. J. McDermott, R. S. Mohanakrishnan, A. N. Naik, J. B. Neaton, S. M. Parmar, K. A. Persson, G. Petretto, T. A. R. Purcell, F. Ricci, B. Rich, J. Riebesell, G.-M. Rignanese, A. S. Rosen, M. Scheffler, J. Schmidt, J.-X. Shen, A. Sobolev, R. Sundararaman, C. Tezak, V. Trinet, J. B. Varley, D. Vigil-Fowler, D. Wang, D. Waroquiers, M. Wen, H. Yang, H. Zheng, J. Zheng, Z. Zhu and A. Jain, *Digital Discovery*, 2025, 4, 1944–1973.
- 3 J. Janssen, S. Surendralal, Y. Lysogorskiy, M. Todorova, T. Hickel, R. Drautz and J. Neugebauer, *Comput. Mater. Sci.*, 2019, 163, 24–36.
- 4 S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko and D. Morgan, *Comput. Mater. Sci.*, 2012, 58, 218–226.
- 5 Y. Zou, A. H. Cheng, A. Aldossary, J. Bai, S. X. Leong, J. A. Campos-Gonzalez-Angulo, C. Choi, C. T. Ser, G. Tom, A. Wang, Z. Zhang, I. Yakavets, H. Hao, C. Crebolder, V. Bernales and A. Aspuru-Guzik, *Matter*, 2025, 8, 102263.
- 6 Z. Wang, H. Huang, H. Zhao, C. Xu, S. Zhu, J. Janssen and V. Viswanathan, DREAMS: Density Functional Theory Based Research Engine for Agentic Materials Simulation, *arXiv*, 2025, preprint, arXiv:2507.14267, DOI: [10.48550/arXiv.2507.14267](https://arxiv.org/abs/2507.14267), <https://arxiv.org/abs/2507.14267>.
- 7 J. Liu, T. Zhu, C. Ye, Z. Fang, H. Weng and Q. Wu, *Chin. Phys. B*, 2025, 34, 117106.
- 8 P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari and R. M. Wentzcovitch, *J. Phys.: Condens. Matter*, 2009, 21, 395502.



- 9 K. Choudhary and F. Tavazza, *Comput. Mater. Sci.*, 2019, **161**, 300–308.
- 10 J. Janssen, E. Makarov, T. Hickel, A. V. Shapeev and J. Neugebauer, *npj Comput. Mater.*, 2024, **10**, 263.
- 11 E. Bosoni, L. Beal, M. Bercx, P. Blaha, S. Blügel, J. Bröder, M. Callsen, S. Cottenier, A. Degomme, V. Dikan, K. Eimre, E. Flage-Larsen, M. Fornari, A. Garcia, L. Genovese, M. Giantomassi, S. P. Huber, H. Janssen, G. Kastlunger, M. Krack, G. Kresse, T. D. Kühne, K. Lejaeghere, G. K. H. Madsen, M. Marsman, N. Marzari, G. Michalick, H. Mirhosseini, T. M. A. Müller, G. Petretto, C. J. Pickard, S. Poncé, G.-M. Rignanese, O. Rubel, T. Ruh, M. Sluydts, D. E. P. Vanpoucke, S. Vijay, M. Wolloch, D. Wortmann, A. V. Yakutovich, J. Yu, A. Zadoks, B. Zhu and G. Pizzi, *Nat. Rev. Phys.*, 2024, **6**, 45–58.
- 12 K. F. Garrity, J. W. Bennett, K. M. Rabe and D. Vanderbilt, *Comput. Mater. Sci.*, 2014, **81**, 446–452.
- 13 M. Schlipf and F. Gygi, *Comput. Phys. Commun.*, 2015, **196**, 36–44.
- 14 A. Willand, Y. O. Kvashnin, L. Genovese, Á. Vázquez-Mayagoitia, A. K. Deb, A. Sadeghi, T. Deutsch and S. Goedecker, *J. Chem. Phys.*, 2013, **138**, 104109.
- 15 E. Kucukbenli, M. Monni, B. I. Adetunji, X. Ge, G. A. Adebayo, N. Marzari, S. de Gironcoli and A. D. Corso, Projector augmented-wave and all-electron calculations across the periodic table: a comparison of structural and energetic properties, *arXiv*, 2014, preprint, arXiv:1404.3015, DOI: [10.48550/arXiv.1404.3015](https://doi.org/10.48550/arXiv.1404.3015), <https://arxiv.org/abs/1404.3015>.
- 16 A. Dal Corso, *Comput. Mater. Sci.*, 2014, **95**, 337–350.
- 17 M. Topsakal and R. Wentzcovitch, *Comput. Mater. Sci.*, 2014, **95**, 263–270.
- 18 M. van Setten, M. Giantomassi, E. Bousquet, M. Verstraete, D. Hamann, X. Gonze and G.-M. Rignanese, *Comput. Phys. Commun.*, 2018, **226**, 39–54.
- 19 M. Sachs, S. I. Ivlev, M. Etter, M. Conrad, A. J. Karttunen and F. Kraus, *Inorg. Chem.*, 2021, **60**, 16686–16699.
- 20 D. Vanderbilt, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1990, **41**, 7892–7895.
- 21 P. E. Blöchl, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1994, **50**, 17953–17979.
- 22 D. R. Hamann, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2013, **88**, year.
- 23 S. Goedecker, M. Teter and J. Hutter, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1996, **54**, 1703–1710.
- 24 G. de Miranda Nascimento, F. J. dos Santos, M. Bercx, D. Grassano, G. Pizzi and N. Marzari, Accurate and efficient protocols for high-throughput first-principles materials simulations, *arXiv*, 2025, preprint, arXiv:2504.03962, DOI: [10.48550/arXiv.2504.03962](https://doi.org/10.48550/arXiv.2504.03962), <https://arxiv.org/abs/2504.03962>.
- 25 S. P. Huber, M. Minotakis, M. Bercx, T. Reents, K. Eimre, N. Paulish, N. Hörmann, M. Uhrin, N. Marzari and G. Pizzi, *Digital Discovery*, 2026, **5**, 1114–1131.
- 26 M. Uhrin, S. P. Huber, J. Yu, N. Marzari and G. Pizzi, *Comput. Mater. Sci.*, 2021, **187**, 110086.
- 27 S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson and G. Ceder, *Comput. Mater. Sci.*, 2013, **68**, 314–319.
- 28 A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder and K. A. Persson, *APL Mater.*, 2013, **1**, 011002.
- 29 G. Prandini, A. Marrazzo, I. E. Castelli, N. Mounet and N. Marzari, *npj Comput. Mater.*, 2018, **4**, 72.
- 30 L. Talirz, S. Kumbhar, E. Passaro, A. V. Yakutovich, V. Granata, F. Gargiulo, M. Borelli, M. Uhrin, S. P. Huber, S. Zoupanos, C. S. Adorf, C. W. Andersen, O. Schütt, C. A. Pignedoli, D. Passerone, J. VandeVondele, T. C. Schulthess, B. Smit, G. Pizzi and N. Marzari, *Sci. Data*, 2020, **7**, 299.
- 31 L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.
- 32 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
- 33 J. H. Friedman, *Ann. Math. Stat.*, 2001, **29**, 1189–1232.
- 34 L. Ward, A. Agrawal, A. Choudhary and C. Wolverton, *npj Comput. Mater.*, 2016, **2**, 16028.
- 35 L. Ward, A. Dunn, A. Faghaninia, N. E. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, K. Chard, M. Asta, K. A. Persson, G. J. Snyder, I. Foster and A. Jain, *Comput. Mater. Sci.*, 2018, **152**, 60–69.
- 36 S. De, A. P. Bartók, G. Csányi and M. Ceriotti, *Phys. Chem. Chem. Phys.*, 2016, **18**, 13754–13769.
- 37 L. Himanen, M. O. Jäger, E. V. Morooka, F. Federici Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke and A. S. Foster, *Comput. Phys. Commun.*, 2020, **247**, 106949.
- 38 A. Y.-T. Wang, S. K. Kauwe, R. J. Murrdoch and T. D. Sparks, *npj Comput. Mater.*, 2021, **7**, 77.
- 39 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, *Adv. Neural Inf. Process. Syst.*, 2017, 5998–6008.
- 40 V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder and A. Jain, *Nature*, 2019, **571**, 95–98.
- 41 T. Xie and J. C. Grossman, *Phys. Rev. Lett.*, 2018, **120**, 145301.
- 42 K. Choudhary and B. DeCost, *npj Comput. Mater.*, 2021, **7**, 185.
- 43 S. Gong, K. Yan, T. Xie, Y. Shao-Horn, R. Gomez-Bombarelli, S. Ji and J. C. Grossman, *Sci. Adv.*, 2023, **9**, eadi3245.
- 44 Y. Romano, E. Patterson and E. Candes, *Adv. Neural Inf. Process. Syst.*, 2019, 3543–3553.
- 45 N. Meinshausen, *J. Mach. Learn. Res.*, 2006, **7**, 983–999.
- 46 C. Molnar, *Interpretable Machine Learning*, 3rd edn, 2025.

