Detection of topological materials with machine learning

Nikolas Claussen,[1] B. Andrei Bernevig,[2] and Nicolas Regnault[1,2]

[1] *Laboratoire de Physique de l'École Normale Supérieure, ENS, Université PSL, Centre National de la Recherche Scientifique, Sorbonne Université, Université Paris-Diderot, Sorbonne Paris Cité, Paris, France*

[2] *Joseph Henry Laboratories and Department of Physics, Princeton University, Princeton, New Jersey 08544, USA*

(Received 11 December 2019; accepted 25 March 2020; published 3 June 2020)

Databases compiled using *ab initio* and symmetry-based calculations now contain tens of thousands of topological insulators and topological semimetals. This makes the application of modern machine learning methods to topological materials possible. Using gradient boosted trees, we show how to construct a machine learning model which can predict the topology of a given existent material with an accuracy of 90%. Such predictions are orders of magnitude faster than actual *ab initio* calculations. We use machine learning models to probe how different material properties affect topological features. Notably, we observe that topology is mostly determined by the "coarse-grained" chemical composition and crystal symmetry and depends little on the particular positions of atoms in the crystal lattice. We identify the sources of our model's errors and we discuss approaches to overcome them.

DOI: 10.1103/PhysRevB.101.245117

I. INTRODUCTION

Topological insulators (TIs) and topological semimetals (TSMs) are solid-state systems which exhibit robust edge or surface modes and quantized bulk response functions due to the topological properties of their electronic wave functions [1,2]. Since the first prediction of TIs in the form of the two-dimensional quantum spin Hall effect [3–5], the field of topological materials has seen remarkable advances. Recently introduced theoretical methods [6–8] have allowed for the large-scale discovery of topological materials.

In particular, the theory of topological quantum chemistry [6] (TQC) provides a unified framework for the treatment of all topological phases arising from crystalline symmetries. TQC relies on the notion of elementary band representations (EBRs), which enumerate a basis for all electronic bands induced from atomic orbitals (atomic limits), and compatibility relations, constraining how bands can connect across the Brillouin zone.

A set of valence bands which cannot be decomposed as a sum of EBRs makes a material into a strong TI. A TSM is characterized by a set of bands below the Fermi level which does not satisfy the compatibility relations and can therefore not be separated from other bands. While TQC can also fully describe so-called fragile topology [9,10] (which can be removed by the addition of topologically trivial bands) and polarization topology [11] (as occurs in the celebrated Su-Schrieffer-Heeger model [12]), in this paper we will focus on strong topology protected by crystal symmetries and dictated by the symmetry eigenvalues at high-symmetry points in the Brillouin zone. Strong topology is, for the purposes of this paper, defined as the topology that is stable to the addition of any atomic limits.

Applying the theory of TQC to the output of *ab initio* calculations, typically density functional theory [13,14] (DFT), on databases of experimentally determined and/or theoretically predicted crystal structures in an automated fashion, large catalogs of topological materials [15–18] have been compiled. These efforts showed that, far from an isolated phenomenon, topology is ubiquitous: At least 30–40% of known stoichiometric materials have some nontrivial topological features [15].

The availability of these datasets, containing tens of thousands of materials, opens the door to modern machine learning (ML) methods [19]. The hope is to bypass the complex, multistep computation necessary to determine the topology of a given material by an empirical, statistical model. In particular, such a model could potentially predict the topological features of a material several orders of magnitude faster. Furthermore, a ML model can infer which quantities decide the topology of a material and possibly offer hints on their role, while *ab initio* calculations are very hard to interpret.

In the present paper, we show how to construct a ML model which can predict the DFT-computed topology of a given material with an accuracy of almost 90%, based on the dataset of Ref. [20]. Our paper is part of the existing body of research on replacing or accelerating DFT by ML methods [21,22]. Yet, it differs from most of this literature, as we do not search to predict "energetic" quantities like the formation energy or the bulk modulus, but an intrinsically quantum property derived from the wave function.

We provide online (see Ref. [23]) a fast and efficient tool to predict the possible topological nature of a given material. Beyond the ML model itself, our results include an analysis of various crystal properties with respect to their relevance for the prediction of the electronic topology. We use machine learning models to probe how and how strongly different material properties affect topological features. We find that the model performance saturates at a small number of properties the impact of which on the topology we show via a simplified model. In particular, information about the positions of atoms in the crystal lattice does not allow for better predictions, within the limitations of our type of ML model.

This paper falls under the ML paradigm known as *supervised learning* [19]. The goal is to infer a functional relationship $f : x \mapsto y$ based on a collection of N examples $\{(x_i, y_i = f(x_i))\}_{i=1,\ldots,N}$ which form a so-called sample. A member i of the sample is called a sample point. Among a class of functions, we numerically search for the one which bears the greatest resemblance to f when evaluated on the sample, as measured by the so-called loss function. This procedure is called training. What sets ML apart from traditional fitting are the large number sample size N and efficient numerical training algorithms, permitting much more complex classes of models.

This paper is organized as follows: First, we discuss in more detail the dataset on which we rely and the ML methodology which we apply. Next, we address the unique challenges which crystal structures pose in the context of ML. We present our model for the prediction of a material's topology and analyze its performance according to different metrics. We use a simplified model to shed light on how the ML model arrives at its predictions and analyze the reasons behind the model's errors. Finally, we highlight obstacles to more sophisticated approaches and discuss our results.

II. RESULTS

A. Dataset

This paper is based on a large catalog of topological materials which was compiled in Ref. [20] and which is accessible online (see Ref. [24]). First-principles calculations, using DFT as implemented in the Vienna Ab-Initio Simulation Package [25], were carried out for 70020 impurity free, crystalline materials selected from the Inorganic Crystal Structure Database (ICSD) [26]. This analysis used nearly 20×10^6 CPU hours. Applying the theory of TQC to the results of the *ab initio* calculation, materials were grouped into five broad categories [20]: insulators with trivial topology (49.5% of the materials in the database); trivial insulators, also referred to as linear combination of EBRs (LCEBR), two types of TIs called Not a Linear Combination of EBRs (NLC) (6.5%) and Split EBR (SEBR) type (7%); and two types of TSMs, Enforced Semimetals (ES, 10% of the dataset) and Enforced Semimetal with Fermi Degeneracy (ESFD, 27%). The bands of an NLC-type insulator cannot be written as a linear combination of EBRs or parts of an EBR, whereas in SEBRs there exists an EBR which is energetically split into two bands, of which only one is occupied. A TSM is an ESFD when there is a high-symmetry point degeneracy at the Fermi level and an ES if the degeneracy is away from the high-symmetry points [15].

Each material is described by its stoichiometric formula, a space group (SG), a unit cell, and the positions of the atoms therein. Many ICSD entries are very similar to each other, describing, for example, different measurements (e.g., at different temperatures) of the same material. All in all, the unique combinations of SG and stoichiometric formula make up only 50% of the database—the other materials are "duplicates." In 98% of cases, two materials exhibiting the same stoichiometry and SG also have the same topological classification. Exceptions are mostly due to the instability of DFT calculations (when a small change in the atomic fractional coordinates changes the topological class), or to extreme changes in the environmental conditions (such as ICSD entries describing high-pressure measurements). For the purpose of this paper, we therefore group materials into equivalence classes with the same stoichiometry and space group and select only one representative from each group. Thus, the effective size of our dataset is 35 009 instead of 70 020.

The materials in our dataset cover a wide range of complexity, one measure being the number of atoms per primitive unit cell (ranging from 1 to 60), belonging to 215 of the 230 different space groups and containing 92 different chemical elements. Figure 1 plots the frequency of TIs among compounds containing a given element. Thanks to the large number of materials in the database, we can put heuristics for the chemistry of topological materials [2] on a statistically solid basis.

B. Classification models

In supervised learning, a classification model is trained by minimizing its prediction error on a sample for which the class of each sample point is known [19]. Such a model is often called a *classifier*. We refer to a sample point's class as its *true label*, in contrast to the output of the predictive model, the *predicted label*.

To evaluate a model after training, we test it by predicting the labels for a second sample or *test set* which was not used during the training process. A detailed description of our testing procedure is given in Appendix A. In particular, the scores reported in the body of this paper are computed using tenfold cross validation, where the sample is repeatedly (a total of ten times) split into testing and training sets in different ways. Averaging over the scores eliminates the arbitrariness in the choice of the test set.

One measure of the quality of predictions on a given test set is the accuracy, the fraction of sample points which are classified correctly. When the dataset is imbalanced (i.e., classes are not equally frequent) the accuracy can, however, be misleading. It should be be compared to a naive baseline model which always predicts the most frequent class—in our case, the baseline accuracy is 49.5%, i.e., the percentage of trivial insulators. To define additional measures, consider a binary classifier (which distinguishes only two labels, positive/negative). For binary prediction, four cases can occur, as illustrated in Table I.

One defines the following quantities:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives } + \text{False Positives}} \tag{1}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives } + \text{False Negatives}} \tag{2}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Precision measures the reliability of a classifier's positive predictions and *recall* measures its ability to find all the true positive sample points. The F_1 score is the harmonic mean of these two quantities.

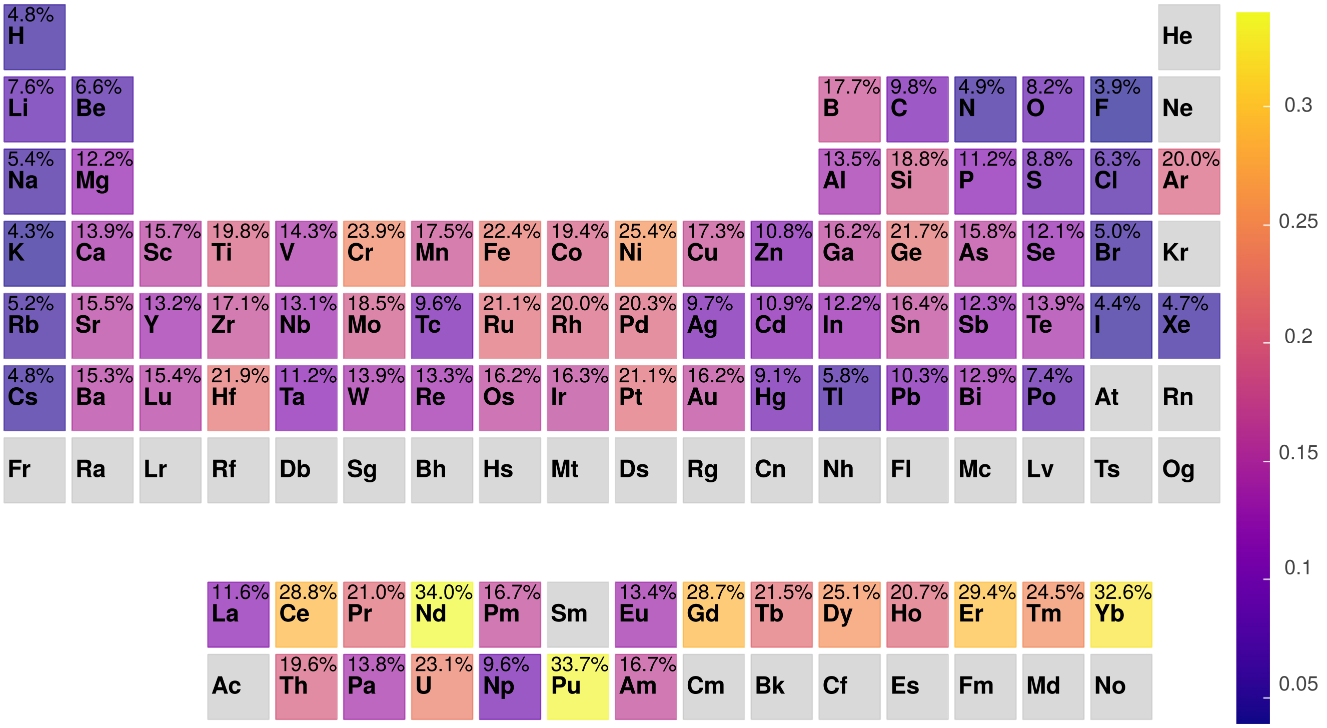

FIG. 1. Chemical composition of TIs (both NLC and SEBR classes). Displayed is the frequency of TIs among compounds containing a given element, which can be interpreted as the conditional probability of a material to be a TI, given that it contains a certain element (the probability is provided by the color scale above). This figure shows that not only heavy elements with strong spin-orbit coupling are capable of forming TIs. Elements forming ionic compounds rarely yield TIs, as their conduction and valence band are formed from different atomic orbitals, making band inversion [5] unlikely. The alternating pattern among the rare earths could be due to the possibility/impossibility of fully filled f-electron bands whenever the number of f electrons is even/odd. However, as f electrons do not typically form dispersive bands [27], this is likely an artifact of DFT.

If a classifier outputs the probability to belong to a class instead of a mere label, precision can be traded for recall and vice versa by changing the threshold probability after which a sample is classified as positive, yielding the precision-recall curve. When there are more than two classes, there is a precision, recall, and F_1 score for each class, characterizing a model's ability to distinguish a specific class from all others. In this paper, we largely used the F_1 score as it provides a single score, largely independent of the choice of threshold, making the comparison between two models straightforward.

In this paper, the models are constructed using the gradient boosted trees (GBTs) algorithm [28] which is based on decision trees (see Fig. 2 for an example). In the ML community, GBT is one of the preferred tools for "tabular" datasets (those with no underlying spatial or temporal structure, in contrast to images or sounds) such as ours. GBT needs no underlying metric to measure the distance between two datapoints (in contrast to a nearest-neighbor classifier, for example) and can therefore naturally deal with situations such as ours where no such metric is available. We empirically tested our GBT model against several other common ML classifiers (random forests, k nearest-neighbor classifiers, linear and Gaussian support vector classifiers [29], as well as a fully connected neural network with dropout regularization) and found it to be superior. The best alternative model, a random forest (also based on decision trees), obtains F_1 scores of 91, 67, and 90% for LCEBRs, TIs, and TSMs, respectively (see Table II).

TABLE I. The error or confusion matrix, indicating the different types of errors of a binary classifier.

	Predicted label positive	Predicted label negative
True label positive	True positive	False negative
True label negative	False positive	True negative

GBT is a special case of the *boosting* technique: Instead of training a single, very complex classifier (a so-called strong classifier), boosting linearly combines the predictions of many simple or *weak* ones. Such a combination or *ensemble* [30] of weak classifiers is expected to have a lower tendency to overfit [19]. In the case of GBT, these classifiers are trees. A weak decision tree is one which has only a small number of nodes. When training a GBT model, one begins by constructing a single tree. Trees are then added iteratively to the model. Each new tree is constructed to correct the error of the existing model. The technical implementation of the GBT model is described in Appendix B.

C. Representation of crystals for ML

Standard ML algorithms like the decision trees of Sec. II B operate on real vectors of a fixed dimension d. Each component of such a vector is referred to as a feature. To apply

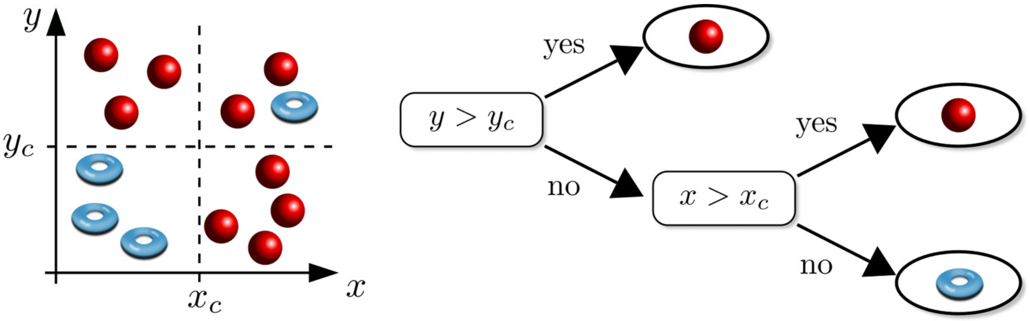

FIG. 2. Example of a decision tree. A decision tree recursively splits the dataset according to a set of binary questions. The depth of the tree is the number of sequential questions it poses (here, two questions), and measures its complexity. In the example, a tree with depth 2 cannot perfectly separate tori from spheres but already provides a fairly good classification. Each node of the tree corresponds to a region in the space (here, spanned by x and y) of input features, the leftmost node corresponding to the entire space. When constructing a decision tree to classify a sample, the tree is built by recursively adding questions in a left-to-right fashion. At each step, the question which maximizes the information gain about the prediction target at each node is chosen.

ML models to the classification of crystalline materials, a map taking crystals to real vectors must be chosen. The choice of this *descriptor* is a nontrivial problem the solution of which is crucial for the success of a predictive model. For example, clearly the descriptor must be independent of the choice of the unit cell and the labeling of a material's constituent atoms. In particular for small sample sizes, the descriptor should also make use of our understanding of chemistry and physics: Elements from the same column in the periodic table are known to be chemically similar. All the while, the dimension d should remain reasonable as the effectiveness of ML models tends to decrease in high dimensions [19]. A large number of descriptors for different applications, e.g., organic chemistry, have been described in the literature [22,31–36]. Recent works [37–39] have also proposed neural network architectures tailored specifically to molecules and crystals.

In this paper we adopt an approach which we dub *single-atom statistics*, which has previously been proposed in Ref. [40] and used, for instance, in Ref. [41] to predict the critical temperature of superconductors. Single-atom statistics is based on a compound's stoichiometric formula. Each of a crystal's constituent atoms is described by a number of physical and chemical single-atom properties, for example, the number of s-shell electrons or the mass. The properties are concatenated to form a vector $\mathbf{X}_i$ for each atom i. To derive the descriptor value for an entire crystal, we calculate statistics over all N_{atom} atoms in a unit cell like the mean or the standard deviation:

$$\text{Mean}[\mathbf{X}] := \frac{1}{N_{\text{atom}}} \sum_{i \in \text{atoms}} \mathbf{X}_i, \tag{4}$$

$$\text{Std}[\mathbf{X}] := \frac{1}{N_{\text{atom}}} \sum_{i \in \text{atoms}} (\mathbf{X}_i - \text{Mean}[\mathbf{X}])^2. \tag{5}$$

The advantage over a direct encoding of a compound's stoichiometric formula is that we can leverage our knowledge of the physically relevant properties. Let us illustrate this by an example. Elements from the same column in the periodic

TABLE II. Performance of GBT models as measured by the F_1 score and accuracy (Acc.). The accuracy should be compared to the baseline of 49.5%. We display the mean and in brackets the standard deviation of tenfold cross-validated scores (see Appendix A). The models differ by the descriptor used to represent materials, defined in column 2. Details on the encoding of N_e and SG are given in Appendix D. d is the total number of properties included in the descriptor.

Model	Descriptor	d	Acc. (%)	F_1 Triv. (%)	F_1 TI (%)	F_1 TSM (%)
Full model (FM)	SG, N_e, *spdf*+, number of atoms from each periodic table row, and column	49	89.7(5)	94.0(3)	70(1)	92.0(5)
FM + Non-SOC	Features used by FM and topological classification of material obtained by DFT without SOC	50	92.0(3)	96.5(2)	77(1)	93.3(4)
Baseline model	SG, N_e, and baseline descriptor (number of atoms from each element in the stoichiometric formula)	94	86.0(5)	92.5(5)	67(1)	91.0(5)
spdf+ model	SG, N_e, and *spdf*+ features	10	87.7(5)	93.0(5)	69(1)	92.0(5)
FM + nearest neighbor	Features used by FM and nearest-neighbor difference features [42], defined in Sec. II C	184	89.0(5)	94.0(3)	69(3)	92.0(5)
FM without SG	N_e, *spdf*+, number of atoms from each periodic table row, and column	48	84.0(5)	91.5(3)	57(2)	86(1)

table are very similar. If we encode the stoichiometric formula directly, replacing an atom in a compound by an atom of a different element of the same column, the encoding vector changes completely. If we use the periodic table column as a descriptor, then the descriptor does not change. Thus, chemically similar substances are represented by nearby vectors, which makes it easier for a machine learning algorithm to infer patterns. Of course, more than one property has to be used, or very different materials might be represented by the same vector.

To validate this approach, we will compare single-atom statistics to a direct encoding of the stoichiometric formula, which we call the *baseline descriptor*. It consists of a 92-dimensional vector. There are 92 different elements in our dataset, and the ith entry equals the fraction of atoms of element i in a compound. For example, NaCl corresponds to the vector $\frac{1}{2}\mathbf{e}_{11} + \frac{1}{2}\mathbf{e}_{17}$ (Na is element number 11, Cl number 17) and, in general, a compound of the form X_mY_n corresponds to the vector $\frac{m}{m+n}\mathbf{e}_X + \frac{n}{m+n}\mathbf{e}_Y$.

In addition to single-atom statistics, we describe crystals by a number of *global* properties like their space group, which pertain to the compound as a whole. Our approach does not take the numeric positions of atoms in the crystal into account.

To decide which single-atom and global properties should be included in our descriptor we began with several hundred physically and chemically motivated quantities, listed in Appendix C. We carried out extensive testing to recursively eliminate irrelevant features. At each testing step, we selected one property and temporarily removed it from the descriptor. If the performance (as measured by the F_1 scores for the three classes trivial, TI, and TSM) did not drop, the property was permanently deleted from the descriptor, else it was retained. Using this procedure we have extracted those features, discussed below, which are relevant for the prediction of a material's topology. A list of the irrelevant features is given in Table VI.

The filtering is essential: Indeed, the inclusion of irrelevant features in the descriptor will both decrease the the quality of a predictive model (due to the chance of spurious correlations between an irrelevant feature and the prediction target) and make its interpretation more difficult. We retained a final test set, not used during the filtering, to verify that the filtering did not lead to overfitting (i.e., that the selected properties work only on the sample used to select them—see Appendix A).

The relevant global properties which we have identified are the SG, deciding which type of topology protected by crystalline symmetry is possible [6], and the number of electrons per unit cell (N_e), deciding whether bands in certain space groups are necessarily completely filled or not. For example, in time-reversal symmetric crystals, the Kramers degeneracy implies that a compound with odd N_e is necessarily a semimetal [15]. Appendix D describes in detail how these properties enter the descriptor map. Among the single-atom properties, the most relevant ones are the mean number of s-, p-, d-, and f-shell valence electrons and the mean and standard deviation [as defined in Eq. (5)] of the atom's row and column number in the periodic table. For example, NaCl has one atom from columns 1 and 31 each, giving a mean column number of 16 and a standard deviation of 15. We will refer to this collection of eight features (mean number of s, p, d, f, mean, and variance of column and row number) as *spdf*+. Finally, the number of atoms in a compound from each column and row in the periodic table is relevant. As an example, half of the atoms in NaCl are in column 1 and 32, respectively, so the column fractions would be represented by the vector $\frac{1}{2}\mathbf{e}_1 + \frac{1}{2}\mathbf{e}_{32}$, where $\mathbf{e}_i$ is the ith unit vector.

To include the numeric positions of atoms in a descriptor we used a technique described in Refs. [42–44]. It calculates differences in single-atom quantities between nearest neighbors, weighted by the boundary area shared by two neighbors (computed using the Voronoi tessellation [44]). An example is the mean difference in electronegativity of neighboring atoms. However, none of these features, to which we refer as "nearest-neighbor differences," proved relevant to our classification problem, as shown by Table II . This finding is consistent with the fact that two ICSD entries with the same stoichiometric formula and SG, but possibly differing atom positions, almost always have the same topological class.

D. Performance of the GBT model

With the features described in Sec. II C and the ML models of Sec. II B we are able to build a classifier which can predict the topology of a given material with an accuracy of 90%, compared to a baseline of 49.5%.

This is demonstrated by the first line of Table II, showing the F_1 score for the different topological classes, as well as the precision-recall curve of Fig. 3. Notably, the algorithm performs remarkably well on TSMs and on trivial insulators, while being less efficient for TIs. We attribute this difference to both the much lower frequency of TIs in the dataset and the more subtle nature of their topology. For instance, whether a material has inverted bands is more difficult to infer from the stoichiometric formula than whether that material has a large gap due to ionic bonds.

We can also configure the model to predict the topological subclass (NLC, SEBR, ES, and ESFD). Table III shows the resulting F_1 scores. Overall, the scores are lower: With more classes come lower sample sizes for each class and more opportunities for misclassification, and thus lower performance. For example, a NLC being classified as a SEBR would not count as error when simply predicting whether a material is a TI.

Some conclusions can be drawn. ES semimetals are harder to detect than ESFD semimetals because the latter can in many cases be detected from N_e and the SG alone. We see that our model reproduces these rules. Likewise, our model correctly identifies the groups in which TIs are allowed according to TQC. Regarding the TIs, NLCs are easier to predict than SEBRs. We attribute this to the more pronounced dependence of SEBR-type topology on energetics, which is not fully captured by our model.

Additionally, we can include partial information from *ab initio* calculations in the model. As most TIs are the result of band inversion driven by spin-orbit coupling (SOC) [2], DFT must be performed with SOC to detect topological materials. However, as Table II shows, already the results of

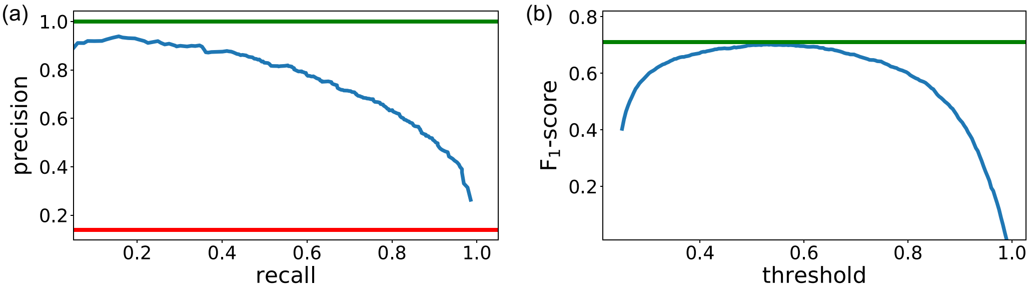

FIG. 3. (a) Precision-recall curve II and (b) F_1 score as a function of the classification threshold, both for the full GBT model from Table II. We focus here on the detection of TIs, the topological class which is both the most interesting and the hardest to detect. Green lines indicate the performance of a perfect model, where high precision and high recall can be achieved at the same time (left), respectively, where precision and recall can be tuned without an overall loss of performance (right). The red line in the left panel shows the null model, which randomly assigns labels according to the threshold probability (the precision is then equal to the overall frequency of TIs in the dataset; see Appendix E). These plots show that over a wide range recall and precision can be exchanged without affecting the overall performance as measured by the F_1 score. A model tuned to have a high precision could, for example, help find promising candidate materials in a database, to be verified by DFT. The left graph shows that at a precision of 90% we have 30% recall.

much less expensive calculations without SOC help our model significantly.

E. Analysis of the GBT model

Let us first comment on the features chosen by our filtering procedure to describe a crystal. Table II shows that a model that only uses the mean number of $s, p, d,$ and f valence electrons, the mean and variance of the periodic table column and row, as well as SG and electron number can capture most of the performance of our full model. It also outperforms the baseline model, which directly translates the chemical formula into a vector. This validates the single-atom statistics approach (see Sec. II C).

The important role of global properties like the SG is shown by the pronounced drop in performance when these features are left out. Finally, we also included much richer collections of properties, the nearest-neighbor differences defined in Sec. II C, which take into account the positions of atoms in the unit cell. Table II shows that they actually perform slightly worse as the inclusion of irrelevant features in a descriptor increases the risk of overfitting.

Let us emphasize that when a feature does not increase the performance of our predictive model we cannot conclude that it is irrelevant for a material's topology—this could simply be due to the failure of the particular ML approach. Yet, the model performance without a certain feature gives an upper bound on the potential feature's influence on the prediction target. For instance, suppose the accuracy of a model on some sample is a%. We can then conclude that the features *not* available to the model only influence the label or class of $(100 - a)$% of the sample.

The other way around, the relevance of some quantity for the model does not imply that it is physically decisive. Rather, it could be only correlated with a physically relevant quantity. However, we know that the input of our model is causally related to the topological class (for instance, TQC shows how changing a material's SG by breaking a crystal symmetry will change the topology), ruling out a mere correlation.

Therefore, we believe that we can draw the following conclusions. Given the space group, the positions of a compound's atoms within the crystal lattice are of limited importance for the topology of a material. Rather, it is the "average orbital character," the mean number of s-, p-, d-, and f-shell valence electrons which counts. Furthermore, we showed the central importance of a material's SG. This hints towards the degrees of freedom that a minimal phenomenological model should take into account.

Next, we want to understand how our model makes use of these features. ML models are often treated as "black boxes": the complexity of their inner workings which makes their high performance possible obfuscates how they arrive at predictions. In the case of a GBT model, the large number of trees which make up the model obstructs its analysis. To address this, we have developed a simplified model which uses only a single tree, depicted in Fig. 4. Despite its simplicity, it performs surprisingly well, with F_1 scores of 84% (trivial), 48% (TI), and 80% (TSM). The tree illustrates explicitly how the GBT model can arrive at its decisions and provides quantitatively supported heuristics for finding topological insulators.

We can also obtain some interesting physical insight from this simplified tree. It shows that a large number of d- or f-shell electrons help to turn a material into a TI. A necessary precondition is a SG which not only theoretically permits but favors (crystalline) topological over trivial insulators. Typical examples are many SGs within the point group D_{2h} (and to a lesser degree D_{4h}), for instance, SG 55.

TABLE III. Performance of the FM when predicting the topological subclass, as measured by the F_1 score and accuracy. The accuracy should be compared to the baseline of 49.5%.

Total accuracy (%)	F_1 Triv. (%)	F_1 NLC (%)	F_1 SEBR (%)	F_1 ES (%)	F_1 ESFD (%)
87.0(3)	94.0(4)	66(2)	59(3)	73(2)	95.5(3)

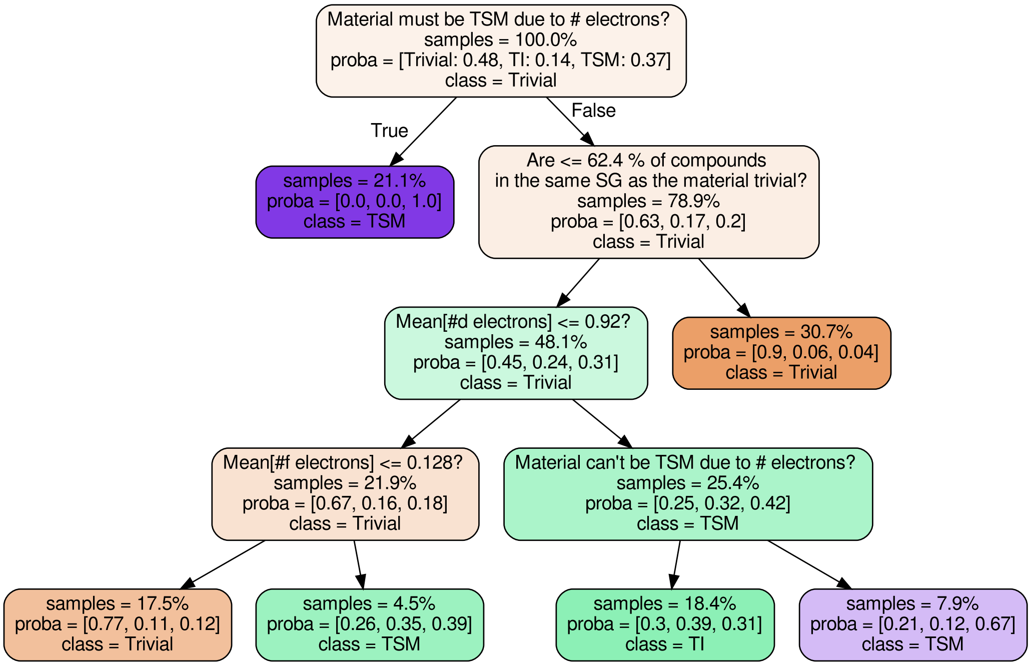
Material must be TSM due to # electrons?
samples = 100.0%
proba = [Trivial: 0.48, TI: 0.14, TSM: 0.37]
class = Trivial
True
False
samples = 21.1%
proba = [0.0, 0.0, 1.0]
class = TSM
Are <= 62.4 % of compounds
in the same SG as the material trivial?
samples = 78.9%
proba = [0.63, 0.17, 0.2]
class = Trivial
Mean[#d electrons] <= 0.92?
samples = 48.1%
proba = [0.45, 0.24, 0.31]
class = Trivial
samples = 30.7%
proba = [0.9, 0.06, 0.04]
class = Trivial
Mean[#f electrons] <= 0.128?
samples = 21.9%
proba = [0.67, 0.16, 0.18]
class = Trivial
Material can't be TSM due to # electrons?
samples = 25.4%
proba = [0.25, 0.32, 0.42]
class = TSM
samples = 17.5%
proba = [0.77, 0.11, 0.12]
class = Trivial
samples = 4.5%
proba = [0.26, 0.35, 0.39]
class = TSM
samples = 18.4%
proba = [0.3, 0.39, 0.31]
class = TI
samples = 7.9%
proba = [0.21, 0.12, 0.67]
class = TSM

FIG. 4. Simplified tree model. The descriptor used is that of the *spdf*+ model (see Table II). To predict the topology of a given material, these features are first calculated. The tree then asks a series of yes/no questions. To understand the graph, consider the top node. Recall from Fig. 2 that each node of the tree corresponds to a subset of the dataset, the top node corresponding to the entire dataset. The "proba" field gives the probability to belong to class trivial/TI/TSM and the "class" field states the most probable one. At the top node, the probabilities are equal to the frequencies of the classes in the entire dataset and the "samples" field, which gives the proportion of the dataset which arrives at a node, is equal to 100%. The top node asks whether the material has to be a TSM due to its SG and N_e. Depending on the answer, a material is either classified as TSM or further questions are asked. The next question, for instance, considers the SG and asks whether the material belongs to a SG the members of which are mostly trivial insulators. While our full model has no input on which combinations of SG and N_e lead to TSMs and learned those rules by itself, for this simplified tree we computed the answer to the question "Material must be TSM due to no. of electrons?" and "Material cannot be TSM due to no. of electrons?" for each material in advance and used it as an input feature. This allows us to keep the tree at a readable size: the tree does not need to ask numerous questions on N_e and SG to answer these questions by itself.

As expected, we recover that TSMs are mostly determined by N_e and SG (due to the "enforced semi-metal" property [6]). See the supplementary material of Ref. [15] for a table indicating which combinations of SG and N_e lead to a TSM.

F. Limitations of the GBT model

The main limitation of any ML model is the number of samples available for training. We show in Fig. 5 how the performance of the model scales with the training set size. Our dataset contains a total of approximately 2500 SEBR- and NLC-type TIs each. This modest sample size is a crucial roadblock. In addition, the materials in the database are very diverse, belonging to many different SGs and containing many different chemical elements. Therefore, for a given topological material, there are often only a few other similar ones, making it harder to infer patterns.

Furthermore, we know that the DFT calculations used to determine the topological labels on which we rely can be unreliable. In such cases, the label may be somewhat random and therefore essentially impossible to predict. Based on a number of criteria (detailed in Appendix F), we selected a subset of 4009 materials for which we are rather confident of the DFT predictions. We will refer to them as *high-confidence* materials. In particular, we selected TIs with high gaps and excluded magnetic and f-electron materials. The results are displayed in Table IV. With respect to Table III, we see significant improvements, which supports the hypothesis that some of the model's errors are due to randomness in the DFT-calculated labels. From a point of view of the search for real-world (not just DFT-calculated) topological materials, it is encouraging to see that large-gap TIs are correctly identified more often.

We can also pinpoint specific types of materials where our model performs poorly. Figure 6 shows that the error rate of the model strongly depends on the symmetry properties of the materials: For cubic point groups, as well as for the hexagonal point group D_{6h}, the model seems to detect topological insulators much less accurately. This is consistent with the fact that these point groups are more complicated in the sense that they contain the greatest number of distinct symmetry operations of all point groups [45].

The error rate also depends strongly on the chemical composition: For materials which contain alkali metals or halogens, the recall for TI is lower than average. This is unsurprising, as the ionic crystals typically formed by these elements are not typically TIs (see Fig. 1).

TABLE IV. Performance of the FM evaluated exclusively on high-confidence materials. Here, we predict the topological subclass and measured the F_1 scores and accuracy. Details on the selection and testing procedure are given in Appendix F.

Total accuracy (%)	F_1 Triv. (%)	F_1 NLC (%)	F_1 SEBR (%)	F_1 ES (%)	F_1 ESFD (%)
92.0(5)	96.0(4)	72(3)	68(1)	79(1)	99.0(5)

III. DISCUSSION

In Ref. [20], the authors compiled the largest-yet catalog of topological materials. Based on this work, we have investigated the prediction of topological classes with machine learning. We showed that a simple and robust ML model, based on gradient boosted trees, can predict the topology of a material's electronic structure with high accuracy, using only its chemical composition and symmetry properties, without costly DFT calculations. Since our ML model cannot guarantee that a real material has topological features, our approach does not claim to replace a full fledged DFT calculation. Still, it provides a fast and efficient tool to predict the possible topological nature of a given material. We also caution that a ML model is only as good as the dataset it was trained on: In spite of its success [15–18], there is no theoretical guarantee that DFT always correctly captures the topological class, which is derived from the electronic wave function. DFT, however, principally models functions of the electron density. We address this issue in Appendix F by considering a subset of materials where we are highly confident in the DFT predictions. Our ML model and the code used to construct it are available online (see Ref. [23]).

The main result of this paper—beyond the predictive model itself—is the use of ML models to probe the relevance of different properties of materials to their topology. We find that our algorithm performance saturates at a very small number of input features, which we identified. Notably, we observe that topology is mostly determined by "coarse-grained" chemical composition and crystal symmetry and depends little on the particular positions of atoms in the crystal lattice. We constructed a simplified model, where the impact of these features can be seen directly. In particular, materials with a large number of d- or f-shell valence electrons in general, and not only compounds containing heavy elements with strong SOC, are likely to be topological insulators.

We investigated several approaches to overcome the limitations and improve the predictions of our model by including the atomic positions in our model, but without success, for instance, the following. TQC emphasizes the decisive role of the symmetry properties of the particular point in the crystal lattice on which an orbital is centered, its *Wyckoff position*. Orbitals on certain Wyckoff positions will induce split EBRs (and thus give rise to SEBR-type TIs). We refer to these positions as SEBR-inducing Wyckoff positions (SEBRiWPs). It seems natural and has been conjectured [10] that compounds with atoms on SEBRiWPs will likely have topological bands. Note that while this does not necessarily imply topology at the Fermi level because these bands can have any energy the existence of topological bands does make topology at the Fermi level more likely. However, we find that only 46% of SEBR-type materials have atoms on SEBRiWPs. This number is not significantly higher than for the materials from other topological classes. Unsurprisingly, the information on whether a material has atoms on SEBRiWPs does not help our predictive model. To make sense of this, let us note that an SEBR is not obliged to have atoms at some SEBRiWP—it is only adiabatically equivalent to a crystal which does. Our results suggest that the SEBR wave functions obtained by *ab initio* calculations are perhaps very different from the idealized ones to which they are adiabatically equivalent. This makes it difficult to use the atomic positions in a predictive model.

Future research should try to take into account the spatial structure of crystals. Given the obstacles we encountered, we believe in the need for more sophisticated ML architectures [such as crystal graph convolutional networks (CGNNs) [39]]. Such architectures can operate directly on crystals and allow one to use the numeric positions of atoms without the need for an explicit descriptor (see Appendix C). However, CGNNs suppose that the quantity to be predicted can sensibly be written as a sum of local contributions, which is appropriate for, e.g., the enthalpy of atomization but not necessarily for predicting the topological class. A challenge is therefore the inclusion of global quantities like the crystal symmetries into these architectures. Alternatively, we must leverage physical understanding to "preprocess" the data on a crystal's spatial structure, making it easier for an ML algorithm to infer patterns. One promising candidate is the use of the empty lattice approximation [46], which can allow an estimate of topological features from the lattice constants only.

Note added. Recently, it was brought to our attention that a related work [47] has appeared.

ACKNOWLEDGMENTS

We cordially thank L. Elcoro for his support by providing data from the Bilbao crystallographic sever. We also thank S. Biermann and M. G. Vergniory for helpful remarks and discussions. B.A.B. and N.R. are supported by U.S. Department of Energy Grant No. desc-0016239; National Science Foundation EAGER Grant No. DMR 1643312; Simons Investigator Grants No. 404513, No. ONR N00014-14-1-0330, and No. NSF-MRSECDMR DMR 1420541; Packard Foundation Grant No. 2016-65128; and the Schmidt Fund for Development of Majorama Fermions funded by the Eric and Wendy Schmidt Transformative TechnologyEric and Wendy Schmidt Transformative Technology Fund.

APPENDIX A: CROSS VALIDATION AND TEST SET

Cross validation allows one to eliminate the dependence of the performance estimate on the way the sample is split into testing and training sets [19]. The entire sample, of size N, is randomly partitioned into k subsets of size N/k, called folds. In stratified cross validation one additionally

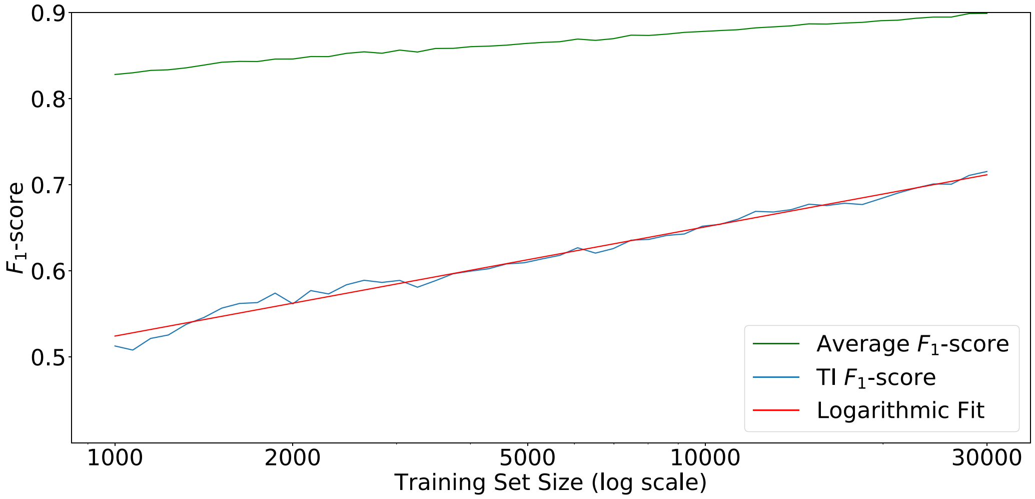

FIG. 5. Dependence of F_1 score on the training set size. The average F_1 score is the average of the F_1 score for the three classes. Performance increases logarithmically with the training set size. A doubling of the training set size increases the F_1 score for TIs by approximately 0.05. Note that a F_1 score is always $\leqslant 1$, so the log dependency cannot hold when the F_1 scores approach this upper bound.

ensures that each fold contains the same proportion of sample points from each class as the entire sample. Now, for each $i \in \{1, \ldots, k\}$ the model is trained using the data from folds $1, \ldots, i-1, i+1, ..k$ and tested on fold i. One can then average over the k scores from the k folds (the fold scores) to get a better performance estimate. We use tenfold stratified cross validation for the estimates stated in Tables II and III. In cross validation the test size is restricted to be equal to N/k for some k, so Fig. 5, which displays the model performance as a function of the training set size, was calculated by an average over five random splits into test and training sets for each point on the graph.

At the beginning of the ML project, we split the dataset into two parts. We used 32 179 materials to develop our GBT model (by analyzing the data and testing different models), and the remaining 2830 materials were used only to test the final model. We refer to them as the final test set. This practice safeguards against overestimating the performance we can achieve by choosing one among several models under investigation (in our case, mostly GBT models using different

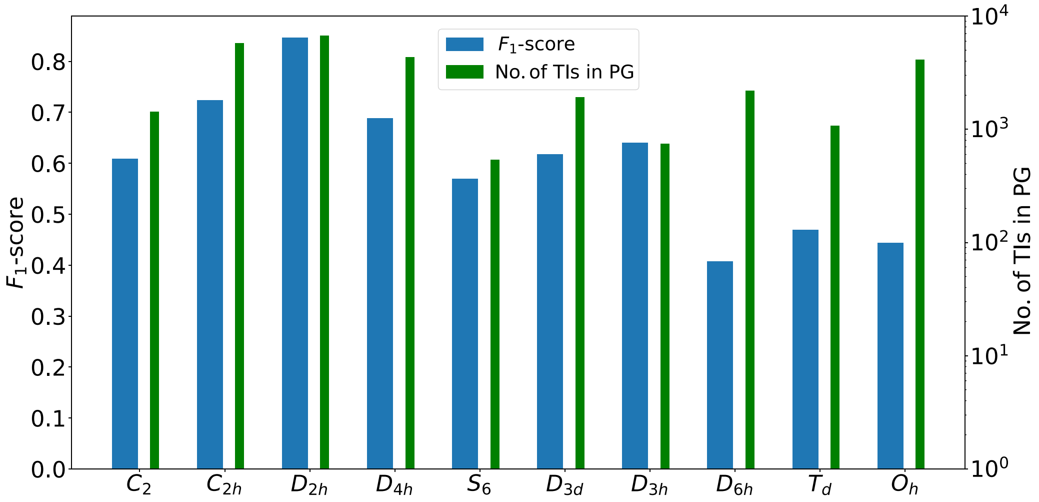

FIG. 6. Dependence of the F_1 score for TIs on the point group (PG). The scores are calculated by testing the model on subsets of the entire test set which contain only materials from a single point group. Here, we use the more coarse-grained PG instead of the SG to obtain larger subsets. Note that the average of the F_1 scores shown in this plot is not equal to the F_1 score on the entire test set: the different subsets contain different proportions of false and correct predictions and the F_1 score is not a linear function of these arguments [see Eq. (3)]. Only PGs which contain any TIs are shown. These are all PGs with an improper rotation (some are not included in the plot, as they do not occur in the dataset frequently enough), a pattern which is correctly learned by the model. Note that for materials from certain PGs (namely, the cubic $\bar{4}3m$ and $m\bar{3}m$ and the hexagonal $6/mmm$) we observe much lower performance. We do not have a clear explanation for this discrepancy. We checked that it cannot be solely explained by different ratios of NLC to SEBR type TIs (the latter are known to be harder to detect) or the differing frequency of TIs in different PGs (see Appendix E).

TABLE V. Hyperparameters for the full GBT model. A full definition of all parameters can be found in Ref. [48].

Parameter	Value	Parameter	Value
Maximal tree depth	10	Minimal child weight	0.1
Learning rate η	0.23	L^2 regularization λ	1.33
Column subsampling by tree	0.78	Column subsampling by node	0.75

features as well as some other ML models investigated before we decided to use GBT).

This can be understood as follows. A collection of many different models $i = 1, \dots, n$ can be thought of as a single "metamodel," in which the index i, i.e., which model to use, is an additional parameter. Optimizing i by choosing the model which performs best on a sample is then a training procedure precisely analogous to optimizing the parameters of a single model. Therefore, overfitting as described in Sec. II B can also occur and an independent final test set is necessary to faithfully evaluate the metamodel performance.

However, because of the limited number of materials in the database and the strong variability of the model error between different materials as discussed in Sec. II F, our final test set is too small to give accurate estimates of the model performance. This is evidenced by the standard deviations of the cross-validated scores given in Table II. We instead checked that the performance on the final test set is within a standard deviation of the mean cross-validated score on the entire dataset of 35 009 materials. For our full model, we find F_1 scores of trivial, 94%; TI, 69%; and TSM, 92%, on the final test set. In spite of minor differences, this agrees to within a standard deviation with the cross-validated scores shown in Table II, showing that we are are not overestimating performance due to model selection.

APPENDIX B: TECHNICAL IMPLEMENTATION

This paper uses the SKLEARN [29] framework for statistical learning for the computation of cross-validated scores and for the training of the single-tree model. The main models are trained using the XGBOOST [48] implementation of gradient boosting. Certain features, for instance, the nearest-neighbor differences, were calculated using MATMINER [44]. The code written for this paper will be made available online (see Ref. [23]).

Hyperparameters

The GBT algorithm controls the complexity of the constructed model with several hyperparameters like the maximal tree depth. Increasing the model complexity increases the tendency to overfit. For instance, in a tree of depth $\log_2(N)$ [in our case, $\log_2(N) \approx 15$], the leaf number (the number of terminal nodes in a tree; in Fig. 2 there are three) equals the sample size N, so that the tree can simply memorize the labels of all training sample points. This will not generalize to new data. If the model complexity is too low, the model cannot fit the data well.

The hyperparameters were chosen through cross-validated random search in the parameter space: We trained models for 300 different randomly chosen values of the hyperparameters, evaluated their performance by cross validation, and chose the model with the best F_1 score. Because there are three different F_1 scores for each model, the "best" model is not uniquely defined, but we found that the three scores are strongly correlated when hyperparameters are varied. So in practice, this problem was not very pressing.

The parameters we chose are listed in Table V. The same parameters were used to compute all scores given in this paper (in order to ensure comparability between models using different descriptors). Of particular importance are the maximal tree depth and the minimal child weight, which controls the minimal number of samples which belong to a tree's leaf (a low value is necessary to deal with SGs and elements which are very rare). Column subsampling randomly selects a given percentage of all features and, when training a new tree, increasing robustness to overfitting. Learning rate and L^2 regularization regularize the weights assigned to individual trees. We furthermore used a weighted loss function to offset the class imbalance. The full model of Table II contains 150 individual trees. Recall that during the training of a GBT model trees are added to the model one after the other. We computed the performance after each such step and found that it saturates after ≈150.

APPENDIX C: TESTED FEATURES AND UNSUCCESSFUL MODELS

In Table VI we provide a list of the features which we tested in the course of developing our model, but which in turn proved irrelevant for the prediction of a material's topological class. In particular, we tested a number of features related to the Wyckoff positions of atoms in the crystal lattice. All the descriptors we tested failed. We believe this to be due to the effect discussed in Sec. III: Wyckoff position related features can give information about the topology if the wave function of a compound is similar to the noninteracting one (i.e., a wave function made out of atomic orbitals centered on the locations of the actual atoms). This is not necessarily the case for the DFT-calculated wave functions underlying the topological class given in our database. However, we cannot exclude that we simply have missed the relevant Wyckoff-related features, due to the large possible variety of such features.

In addition to the features of Table VI which were used together with GBT models, we also experimented with a number of machine learning models different from GBT. In particular, we tried to directly use the CGNN architecture of Ref. [37], which is available as open source software, however without success. CGNNs are similar to the convolutional neural networks which have been immensely successful in image recognition [49]. Crystals are represented by a graph, where each atom from the unit cell is a vertex and two

TABLE VI. Irrelevant, tested features by category. For all single-atom statistics (SAS) features, all nearest-neighbor difference features, and the spatial composition features (except for the compound dimensionality, which is a global feature) we considered the following statistics: mean, variance, minimum, and maximum. For the Wyckoff position features we did not only consider these statistics. We also calculated a weighted average number of s-, p-, d-, and f-shell valence electrons, where each atom is weighted according to its Wyckoff position (for example, according to the multiplicity of the Wyckoff position) to capture the interaction between the symmetry of the atomic orbitals and that of an atom's position in the crystal.

Category	Tested features
Features derived from SG	Bravais lattice, point group, and subgroups of the space group
Unit cell	Lattice parameters, their ratios, angles of lattice vectors, unit-cell volume, density, and volume per atom
Stoichiometric features	L^2 norm of fractions of elements in the material [40] and number of atoms per unit cell
Ionic character	Possibility to form an ionic crystal from common oxidation states of the atoms [44]
HOMO/LUMO energies	Energy and type (s, p, d, or f) of the highest occupied and lowest unoccupied atomic orbital [44]
Chemical features (SAS)	Covalent radius, electronegativity, electron affinity, atomic mass, Mendeleev number [40], melting temperature of the element in pure form, and atomic charge Z^4 (measures strength of SOC coupling)
Valence orbitals (SAS)	Number of unfilled s, p, d, f orbitals
Nearest-neighbor differences	Atomic charge, Mendeleev number, periodic table row, and column, Atomic mass, covalent radius, electronegativity, and number of filled and unfilled s, p, d, and f valence orbitals [42]
Spatial composition	Bond lengths (distances to nearest neighboring atom), bond angles (angles of triples of atoms), dimensionality of the compound, and boundary area between neighbors as obtained by Voronoi tessellation [44]
Wyckoff positions	Multiplicity of Wyckoff position, order of the group of symmetries fixing the position (stabilizer), whether the stabilizer contains inversion, and whether the Wyckoff position induces split EBRs

atoms are joined by an edge if they are next neighbors. The vertex values undergo a series of nonlinear transformations which depend only on the vertex's neighbors and are finally combined in a weighted sum, yielding the prediction. The same transformation is applied to all vertices, making it a graph convolution. Each transformation corresponds to one layer of the neural network, which is trained numerically. This procedure means that no explicit descriptor must be chosen, and the network can operate directly on crystals. However, as mentioned in Sec. III, the CGNN architecture expresses the quantity to be predicted as a sum of local contributions. Such an ansatz might not be appropriate for topological classes.

We also used another CGNN model from Ref. [37] in an unsuccessful attempt of so-called transfer learning [50], trying to overcome the limitations due to the the size of our dataset. Transfer learning can make use of datasets for which the topological class is not known, but related quantities have been computed. The (DFT-calculated) band gap of a material provides useful information on a material's topology (for instance, semimetals are gapless and large gaps of more that 1 eV cannot be topological). Indeed, the TI F_1 score of our model improves from 70 to 75% if we provide the *actual* DFT gap. The idea was to predict the band gaps with the CGNN and then feed this prediction as an additional feature to our GBT model. While this particular attempt did not work (likely because the predicted gaps were not sufficiently accurate), the general strategy, which is related to the ML concept of *stacking* [30], seems promising. In particular, it can make use of datasets for which some relevant quantities like the band gap have been computed, but the topological class has not been computed and combines different descriptors which would be unsuccessful on their own.

Further, we considered some of the descriptors previously discussed in the literature [22,31–36]. A large number of these are not well adapted to samples containing a wide range of chemical elements and many atoms per unit cell because the dimension d of the descriptor becomes very large. We considered, for example, the sine matrix descriptor [51] in combination with a kernel support vector machine classifier [52] (with Gaussian kernel), an approach common in the literature [53]. For a material with N_{atom} atoms with nuclear charges $\{Z_i\}_{i=1,\dots,N_{\text{atom}}}$ at positions $\{\mathbf{R_i}\}_{i=1,\dots,N_{\text{atom}}}$ in the unit cell it is the $N_{\text{atom}} \times N_{\text{atom}}$ matrix defined by

$$C_{ij}^{\text{sine}} = \begin{cases} \frac{Z_i Z_j}{|\mathbf{B}\cdot\sum_{k=\{x,y,z\}} \hat{\mathbf{e}}_k \sin^2\left(\pi \mathbf{B}^{-1}\cdot(\mathbf{R}_i - \mathbf{R}_j)\right)|} & \text{for } i \neq j \\ \frac{Z_i^{2.4}}{2} & \text{for } i = j \end{cases}. \quad \text{(C1)}$$

Here, $\mathbf{B}$ is the 3×3 matrix formed from the lattice basis vectors. Empirically, we find that such a model cannot distinguish different topological classes. The sine matrix was originally conceived as a proxy for the electrostatic force field due to the atomic nuclei. This makes it suitable, for instance, for models which predict mechanical properties like the bulk modulus. In contrast to the former, topological classes have no clear link to interatomic force fields and the failure of this and related descriptors like the Ewald matrix [51] is unsurprising.

Finally, approximately 80% of the materials in the ICSD are classified into so-called material prototypes according to their crystal structure [26]. For example, the NaCl type is defined to contain all materials from SG 225 with two different atoms per unit cell at positions [0,0,0] and $[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$ in the standard basis of lattice vectors (this is the crystal structure of NaCl). We selected the 100 most common material prototypes (a total of 10 389 materials) and encoded the prototype with the one-hot method as defined by Eq. (D2). This approach also failed to improve over the model shown in Table II.

APPENDIX D: ENCODING OF CATEGORICAL FEATURES

The global features SG and N_e we described are *categorical*: they indicate membership in a class, for example, in the materials with a certain SG. This fact should be reflected by the way the feature is encoded as a real vector for the ML algorithm [19]: Most algorithms assume that two samples are similar if the difference of the two corresponding vectors is small. However, e.g., SG 9 contains no crystalline TIs whereas in SG 10 47% of materials are crystalline TIs. Accordingly, the actual number of a SG is not a reasonable encoding by itself—this number is somewhat arbitrary (but not completely: SGs are ordered according to point group and lattice type). We therefore supplement the SG number with an additional descriptor. We calculate the following frequencies for each SG $g \in \{1, \dots, 230\}$ and topological class $i \in \{$ trivial, NLC, SEBR, ES, ESFD$\}$:

$$\frac{\text{Number of materials with top. class } i \text{ and SG } g}{\text{Number of materials with SG } g}. \tag{D1}$$

This can be interpreted as the conditional probabilities to belong to a topological class given the SG. Equation (D1) is then used to encode the SG, e.g., a material with SG 10 would be assigned the vector [0.42 (trivial), 0.47 (NLC), 0 (SEBR), 0 (ES), 0.11 (ESFD), 10 (SG number)].

In particular, when we evaluate a model using a training and a test set, Eq. (D1) is calculated using the materials in the training set only. Otherwise, information about the test set (namely, how topological class and SG are related in the test set) would indirectly be available when the model is trained (this is called target leakage in ML). Note that using *only* Eq. (D1) and not the raw SG number only lowers the TI F_1 score by approximately 1% point.

For similar reasons, N_e is not fed directly into the model but instead is represented by the digits in its binary representation, e.g., $N_e = 7$ would be represented by the vector [1,1,1]. This makes it easy for a decision tree to check, for example, whether N_e is odd. Recall that the leaf of a decision tree corresponds to a connected region in the space of the input vectors (see Fig. 2). So a tree which splits the numbers $1, \dots, m$ into even and odd numbers based on their numerical value needs to split $[1, m]$ into m connected regions (here, intervals) and thus has m leaves. With the binary encoding, a tree with two leaves suffices.

We also tested different ways of encoding the SG. One often-used option to encode categorical features is *one-hot* encoding. For the case of N_c different classes, it is defined by the following mapping:

$$\{1, \dots, N_c\} \to \mathbb{R}^{N_c}, \quad i \mapsto \mathbf{e}_i \tag{D2}$$

where $\mathbf{e}_i$ is the ith unit vector. However, the method described above proved superior.

APPENDIX E: RANDOM CLASSIFIERS

The performance of a classifier depends strongly on the frequency of the different classes in the dataset. Let us consider a binary classifier and a dataset where the fraction of positive samples is q. To understand the dependence of a classifier on q, it is useful to consider a toy example: the random classifier, which assigns a positive label with probability p and a negative one with probability $1 - p$ (a purely positive

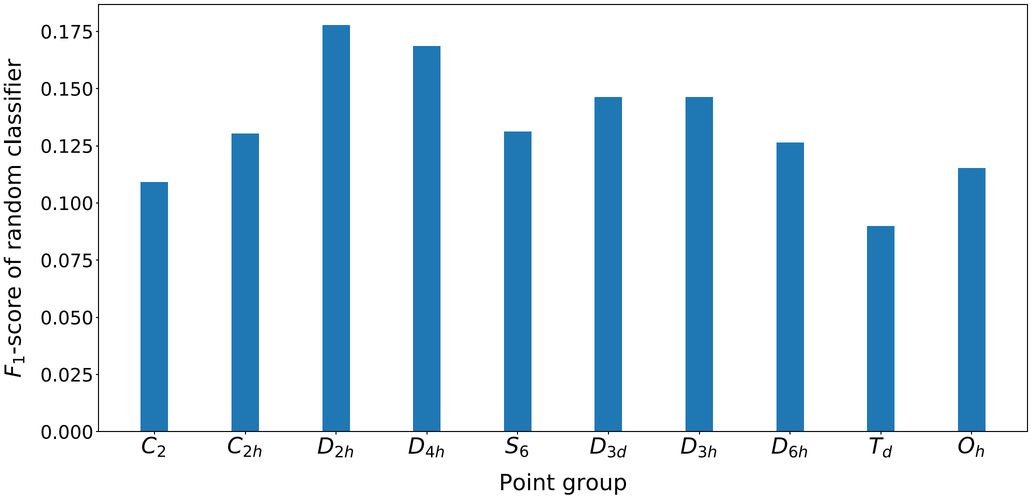

FIG. 7. Dependence of the F_1 score of a random classifier on the the PG. Each PG contains a different fraction q of TIs, so according to Eq. (E2) a random classifier will perform differently on them. While these F_1 scores are of course much lower than those of Fig. 6 (please note that the vertical scale is different), they can explain the variation of the actual scores between the first seven PGs ($\bar{1}$ to $\bar{6}2m$). However, the performance drop observed in Fig. 6 for the last three PGs is much more severe than would be expected from the random classifier model.

classifier is the special case $p = 1$). Now the dependence of various metrics on q is simple to calculate. For the metrics considered in this paper, one has

$$\text{accuracy} = pq + (1-p)(1-q), \tag{E1}$$

$$\text{precision} = q,\ \ \text{recall} = p,\ \ F_1 = 2\frac{pq}{p+q}. \tag{E2}$$

These results can help explain patterns seen in real classifiers and in particular highlight behavior which deviates from these simple models. First, we can judge the quality of our model by comparing it to a random classifier, as done in the precision-recall curve of Fig. 3. Second, the F_1 score of the random classifier can partially explain the behavior seen in Fig. 6. Here, the frequency of positive samples, i.e., TIs, varies between different datasets, i.e., materials from different point groups, while the classifier remains the same. We therefore consider a random classifier with p fixed to 0.14 (the frequency of TIs in the overall dataset). The results are shown in Fig. 7.

APPENDIX F: HIGH-CONFIDENCE TOPOLOGICAL MATERIALS

As discussed in Sec. II F, we believe that a part of the errors of our model is due to unreliability of the DFT calculations. To test this hypothesis, we selected a number of topological materials (according to the criteria detailed below) where we have a high confidence in the predictions of DFT. We refer to these compounds as *high-confidence* topological materials or HC materials.

We then tested the performance of our model exclusively on HC materials, using a modified cross-validation scheme. We split the HC materials into five stratified folds (see Appendix A), trained the model on the union of the non-HC materials and four of the folds, and finally evaluated the model on the remaining fold (the results are displayed in Table IV). This way, we ensure that some HC materials are present in the training set. Otherwise, training and test sets would contain very different materials, which makes it impossible to make an estimate of the model performance.

To select the HC materials, we used criteria based on both properties of the materials and the DFT results. Regarding the first, we exclude f-electron and magnetic materials. Regarding the latter we chose TIs with large direct gaps (>0.025 eV for NLC and >0.035 eV for SEBR, corresponding to the highest 20% of gaps) and TSMs with low numbers (>32 for ES and >25 for ES) of Fermi level crossings. This way, we selected a total of 1978 trivial, 249 NLC, 316 SEBR, 404 ES, and 1090 ESFD materials. The proportions of the topological classes among the HC materials are almost the same as in the entire dataset, to avoid any distortions when testing a model on the HC materials.

[1] M. Z. Hasan and C. L. Kane, Colloquium: Topological insulators, Rev. Mod. Phys. **82**, 3045 (2010).

[2] A. Bansil, H. Lin, and T. Das, Colloquium: Topological band theory, Rev. Mod. Phys. **88**, 021004 (2016).

[3] C. L. Kane and E. J. Mele, Quantum Spin Hall Effect in Graphene, Phys. Rev. Lett. **95**, 226801 (2005).

[4] C. L. Kane and E. J. Mele, Z2 Topological Order and the Quantum Spin Hall Effect, Phys. Rev. Lett. **95**, 146802 (2005).

[5] B. A. Bernevig, T. L. Hughes, and S.-C. Zhang, Quantum spin Hall effect and topological phase transition in HgTe quantum wells, Science **314**, 1757 (2006).

[6] B. Bradlyn, L. Elcoro, J. Cano, M. G. Vergniory, Z. Wang, C. Felser, M. I. Aroyo, and B. A. Bernevig, Topological quantum chemistry, Nature (London) **547**, 298 (2017).

[7] H. C. Po, A. Vishwanath, and H. Watanabe, Symmetry-based indicators of band topology in the 230 space groups, Nat. Commun. **8**, 50 (2017).

[8] Z. Song, T. Zhang, Z. Fang, and C. Fang, Quantitative mappings between symmetry and topology in solids, Nat. Commun. **9**, 3530 (2018).

[9] H. C. Po, H. Watanabe, and A. Vishwanath, Fragile Topology and Wannier Obstructions, Phys. Rev. Lett. **121**, 126402 (2018).

[10] J. Cano, B. Bradlyn, Z. Wang, L. Elcoro, M. G. Vergniory, C. Felser, M. I. Aroyo, and B. A. Bernevig, Building blocks of topological quantum chemistry: Elementary band representations, Phys. Rev. B **97**, 035139 (2018).

[11] W. A. Benalcazar, B. A. Bernevig, and T. L. Hughes, Quantized electric multipole insulators, Science **357**, 61 (2017).

[12] W. P. Su, J. R. Schrieffer, and A. J. Heeger, Solitons in Polyacetylene, Phys. Rev. Lett. **42**, 1698 (1979).

[13] P. Hohenberg and W. Kohn, Inhomogeneous electron gas, Phys. Rev. **136**, B864 (1964).

[14] W. Kohn and L. J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. **140**, A1133 (1965).

[15] M. G. Vergniory, L. Elcoro, C. Felser, N. Regnault, B. A. Bernevig, and Z. Wang, A complete catalogue of high-quality topological materials, Nature (London) **566**, 480 (2019).

[16] T. Zhang, Y. Jiang, Z. Song, H. Huang, Y. He, Z. Fang, H. Weng, and C. Fang, Catalogue of topological electronic materials, Nature (London) **566**, 475 (2019).

[17] F. Tang, H. C. Po, A. Vishwanath, and X. Wan, Comprehensive search for topological materials using symmetry indicators, Nature (London) **566**, 486 (2019).

[18] M. Wieder *et al.* The renaissance era of topological materials discovery, 2019 (unpublished).

[19] Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning* (MIT, Cambridge, MA, 2012).

[20] B. Wieder, L. Elcoro, B. A. Bernevig, N. Regnault, and M. G. Vergniory, A Complete Catalogue of All Topological Materials (unpublished).

[21] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Machine learning for molecular and materials science, Nature (London) **559**, 547 (2018).

[22] G. R. Schleder, A. C. M. Padilha, C. M. Acosta, M. Costa, and A. Fazzio, From DFT to machine learning: Recent approaches

to materials science: A review, J. Phys.: Mater. **2**, 032001 (2019).

[23] See https://www.topologicalquantumchemistry.com/mltqc.

[24] See http://topologicalquantumchemistry.org.

[25] G. Kresse and D. Joubert, From ultrasoft pseudopotentials to the projector augmented-wave method, Phys. Rev. B **59**, 1758 (1999).

[26] I. Levin, *NIST Inorganic Crystal Structure Database (ICSD)* (National Institute of Standards and Technology, 2018).

[27] A. Georges, in *Lectures on the Physics of Highly Correlated Electron Systems VIII: Eighth Training Course in the Physics of Correlated Electron Systems and High-Tc Superconductors*, edited by F. Mancini and A. Avella, AIP Conf. Proc. No. 715 (AIP, New York, 2004), p. 3.

[28] J. H. Friedman, Greedy Function Approximation: A Gradient Boosting Machine, Ann. Stat. **29**, 1189 (2001).

[29] F. Pedregosa *et al.*, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. **12**, 2825 (2011).

[30] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. **33**, 1 (2010).

[31] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning, Phys. Rev. Lett. **108**, 058301 (2012).

[32] A. P. Bartók, R. Kondor, and G. Csányi, On representing chemical environments, Phys. Rev. B **87**, 184115 (2013).

[33] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, How to represent crystal structures for machine learning: Towards fast prediction of electronic properties, Phys. Rev. B **89**, 205118 (2014).

[34] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko, Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space, J. Phys. Chem. Lett. **6**, 2326 (2015).

[35] M. Hirn, S. Mallat, and N. Poilvert, Wavelet scattering regression of quantum chemical energies, Multiscale Model. Simul. **15**, 827 (2017).

[36] H. Huo and M. Rupp, Unified representation of molecules and crystals for machine learning, arXiv:1704.06439 (2017).

[37] T. Xie and J. C. Grossman, Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties, Phys. Rev. Lett. **120**, 145301 (2018).

[38] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Quantum-chemical insights from deep tensor neural networks, Nat. Commun. **8**, 13890 (2017).

[39] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, Neural message passing for quantum chemistry, in *Proceedings of the 34th International Conference on Machine Learning*, edited by D. Precup and Y. W. Teh, Proceedings of Machine Learning Research Vol. 70 (PMLR, International Convention Centre, Sydney, Australia, 2017).

[40] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, A general-purpose machine learning framework for predicting properties of inorganic materials, npj Comput. Mater. **2**, 16028 (2016).

[41] V. Stanev, C. Oses, A. G. Kusne, E. Rodriguez, J. Paglione, S. Curtarolo, and I. Takeuchi, Machine learning modeling of superconducting critical temperature, npj Comput. Mater. **4**, 29 (2018).

[42] L. Ward, R. Liu, A. Krishna, V. I. Hegde, A. Agrawal, A. Choudhary, and C. Wolverton, Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations, Phys. Rev. B **96**, 024104 (2017).

[43] O. Isayev, C. Oses, C. Toher, E. Gossett, S. Curtarolo, and A. Tropsha, Universal fragment descriptors for predicting properties of inorganic crystals, Nat. Commun. **8**, 15679 (2017).

[44] L. Ward *et al.*, Matminer: An open source toolkit for materials data mining, Comput. Mater. Sci. **152**, 60 (2018).

[45] C. J. Bradley and A. P. Cracknell, *The Mathematical Theory of Symmetry in Solids* (Clarendon, Oxford, 1972).

[46] N. W. Ashcroft and N. D. Mermin, *Solid State Physics*. HRW international ed. (Holt, Rinehart and Winston, New York, 1976).

[47] N. Andrejevic, J. Andrejevic, C. H. Rycroft, and M. Li, Machine learning spectral indicators of topology, arXiv:2003.00994 (2020).

[48] T. Chen and C. Guestrin, XGBoost, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2016), pp. 785–794.

[49] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT, Cambridge, MA, 2016).

[50] S. J. Pan and Q. Yang, A Survey on transfer learning, IEEE Trans. Knowl. Data Eng. **22**, 1345 (2010).

[51] F. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, Crystal structure representations for machine learning models of formation energies, Int. J. Quantum Chem. **115**, 1094 (2015).

[52] B. Schölkopf and A. K. Smola, *Learning with Kernels* (MIT, Cambridge, MA, 2002).

[53] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, and O. A. von Lilienfeld, Prediction errors of molecular machine learning models lower than hybrid DFT error, J. Chem. Theory Comput. **13**, 5255 (2017).