JPhys Materials

PAPER • OPEN ACCESS

Simultaneous learning of several materials properties from incomplete databases with multi-task SISSO

IOP Publishing J. Phys.: Mater. **2** (2019) 024002 https://doi.org/10.1088/2515-7639/ab077b

JPhys Materials

OPEN ACCESS

RECEIVED
3 January 2019

REVISED
6 February 2019

ACCEPTED FOR PUBLICATION
15 February 2019

PUBLISHED
14 March 2019

PAPER

Simultaneous learning of several materials properties from incomplete databases with multi-task SISSO

Runhai Ouyang, Emre Ahmetcik, Christian Carbogno, Matthias Scheffler and Luca M Ghiringhelli[1]

Fritz-Haber-Institut der Max-Planck-Gesellschaft, D-14195 Berlin-Dahlem, Germany

[1] Author to whom any correspondence should be addressed.

E-mail: **ouyang@fhi-berlin.mpg.de**, **ahmetcik@fhi-berlin.mpg.de** and **ghiringhelli@fhi-berlin.mpg.de**

Keywords: artificial intelligence, compressed sensing, crystal structure prediction, metal/nonmetal classification

Supplementary material for this article is available online

Abstract

The identification of descriptors of materials properties and functions that capture the underlying physical mechanisms is a critical goal in data-driven materials science. Only such descriptors will enable a trustful and efficient scanning of materials spaces and possibly the discovery of new materials. Recently, the sure-independence screening and sparsifying operator (SISSO) has been introduced and was successfully applied to a number of materials-science problems. SISSO is a compressed sensing based methodology yielding predictive models that are expressed in form of analytical formulas, built from simple physical properties. These formulas are systematically selected from an immense number (billions or more) of candidates. In this work, we describe a powerful extension of the methodology to a 'multi-task learning' approach, which identifies a single descriptor capturing multiple target materials properties at the same time. This approach is specifically suited for a heterogeneous materials database with scarce or partial data, e.g. in which not all properties are reported for all materials in the training set. As showcase examples, we address the construction of materials properties maps for the relative stability of octet-binary compounds, considering several crystal phases simultaneously, and the metal/insulator classification of binary materials distributed over many crystal prototypes.

1. Introduction

The materials-genome initiative [1] inspired the establishment of several high-throughput computational materials-science projects, leading to the creation of worldwide accessible materials databases [2–5]. In this context, the Novel Materials Discovery (NOMAD) Repository and Archive is the biggest data base for input and output files of density-functional theory (DFT) calculations for materials considering all important computer codes of the community [6–8]. It plays synergistically together with other important data bases, in particular AFLOW [2], Materials Project [3], and OQMD [4].

This wealth of available data opens the era of the data-driven materials science [7, 9], which is fueled by the computer-aided analysis of the data, in order to find patterns and trends otherwise invisible to the human eye. This, in turn, may lead to accelerate discoveries of new materials or phenomena.

A key goal of materials science is to find materials with a high performance in several functions, e.g. stability and catalytic activity and selectivity for a very specific chemical reaction. It is important to realize that the number of materials that qualify is typically very small. However, the complexity and intricacy of the actuating processes is significant. Falling under the umbrella names of artificial-intelligence or (big-)data analytics (terms that include data mining, machine/statistical learning, deep learning, compressed sensing (CS), etc), several methods have been developed and applied to existing materials-science data [10–19] in order to predict properties of interest.

The $T = 0$ K properties of materials are fully described by the many-body Hamiltonian, which is uniquely identified by its descriptors: the position and charges of the atomic nuclei $\{R_I, Z_I\}$ and the number of electrons

N^e. Although, in principle, these could be also descriptors for an artificial-intelligence algorithm, their connection with the materials properties and functions is too complicated, indirect, intricate. As a consequence, the description of processes ruling materials properties and functions requires to add as much domain knowledge to the artificial-intelligence step as available. Obviously, if not done with utmost care, this may well yield a biased and unreliable description. From the mentioned 'fundamental primary' descriptors, $\{R_\mathrm{I}, Z_\mathrm{I}\}$ and N^e, it is also clear that there are two types of needed information: (1) the topology of the atomic structure and (2) the electronic/chemical property of the atoms. When geometry changes are not relevant (or trivial) the first aspect can be simplified or even neglected, and when changes in chemical bonding are nor relevant (or trivial), the second aspect can be simplified or even neglected. We will get back to these issues in the specific application examples discussed below.

Following the strategy introduced in [20], the descriptor can be learned from the data, more precisely the best descriptor can be identified among a possibly immense set of candidates by exploiting a signal-analysis technique known as CS [20–24]. Sure-independence screening and sparsifying operator (SISSO) [25] is a recently developed CS-based method, designed for identifying low-dimensional descriptors (a descriptor is defined as a vector of features, so that the number of features is the dimension of the descriptor) for material properties. It is an iterative scheme that combines the sure-independence screening (SIS) [26] scheme for dimensionality reduction of huge features space and the sparsifying operators for finding sparse solutions. SISSO improves the results over conventional CS methods such as the Linear Absolute Shrinkage and Selection Operator (LASSO [27]), or LASSO-based [20, 24] and greedy algorithms [28, 29] when features are correlated, and can efficiently manage immense features spaces. SISSO has been already successfully applied to identifying descriptors for relevant materials-science properties [25, 30, 31].

In this work, we introduce a learning scheme, termed multi-task (MT) SISSO, within the framework of the wider class of learning schemes known as multi-task learning (MTL) [32–39]. A *task* for a learning algorithm is the learning of a target property starting from a single input source (set of features). The learning of *multiple tasks* (or MTL) is an umbrella term that refers to [38] (i) the learning of multiple target properties using a single input source, *or* (ii) the joint learning of a single target property using multiple input sources, or (iii) a mixture of both. The key aspect is the parallel learning of multiple tasks, with the (sometimes implicit) assumption that the shared information among different tasks can lead to better learning performance if all the tasks are learned jointly, as compared to learning them independently. In other words, MTL assumes that the learning of one task can improve the learning of the other tasks [38]. Though MTL has not yet been applied to materials-science problems so far, it has already been widely applied in other fields, such as in the handwriting recognition problem, self-driving automation system, computer vision, bioinformatics and health informatics, speech and language recognition, and more [32, 33, 38, 39].

In order to clarify how the MTL concept can be applied in materials science, let us introduce the showcase examples that will be addressed in the following sections. Arguably one of the fundamental challenge in materials science is predicting the ground-state crystal structure of a material, given its chemical composition. In [20, 24, 25], models for predicting the relative stability or rock-salt (RS) versus zinc-blende (ZB) structures for *AB* octet binaries were learned via a LASSO-based and SISSO algorithm. Learning models for the prediction of the relative stability of more than two crystal structures, given the same set of chemical formulas, can be cast into MTL. Each difference in energy between crystal structures is a *task* and the common input is the chemical formula and/or a list of properties of the atomic species listed in the chemical formula. The joint learning, in the SISSO framework, sets in when the same descriptor is imposed to be selected for all tasks. More specifically, SISSO identifies models in form of linear mappings between the descriptor $\boldsymbol{d}$—a vector of nonlinear functions of physical properties termed *primary features*—and the property of interest $P = \boldsymbol{dc}$, where $\boldsymbol{c}$ is the vector of coefficients that maps $\boldsymbol{d}$ into P. If we now consider a set $\{P^{(1)}, P^{(2)}, \ldots, P^{N^\mathrm{T}}\}$ of N^T properties (e.g. the set of energy differences between crystal structures for the same chemical formula), the idea of MTL applied to SISSO is to find models $P^k = \boldsymbol{d} \cdot \boldsymbol{c}^k$ where the set of fitting coefficients $\{\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{N^\mathrm{T}}\}$ maps *the same descriptor* $\boldsymbol{d}$ into the different properties $\{P^{(1)}, P^{(2)}, \ldots, P^{N^\mathrm{T}}\}$. In section 3.1, we will show the results of such learning. Besides the physical meaningfulness and Occam-razor-reminiscent elegance that a few mechanisms are ruling all energy differences (though with different relative importance), a great advantage of the MTL framework is to allow for a robust learning also when the training database (in this case, reference energy differences) is incomplete, i.e. for several chemical formulas only some of the energy differences are known. As we will show, MT-SISSO learns accurate predictive models also with high levels of incompleteness (e.g. when 50% or more of the information is randomly missing). In figure 1, we show graphically the setup for MT-SISSO, in particular in terms of the possibility to deal with incomplete data.

A second setup where MT-SISSO is helpful is the learning of one common property of many materials belonging to physically different groups, e.g. they have different bonding characteristics and their ground-state crystal structure belong to different space groups. Obviously, in such situation one single predictive model is

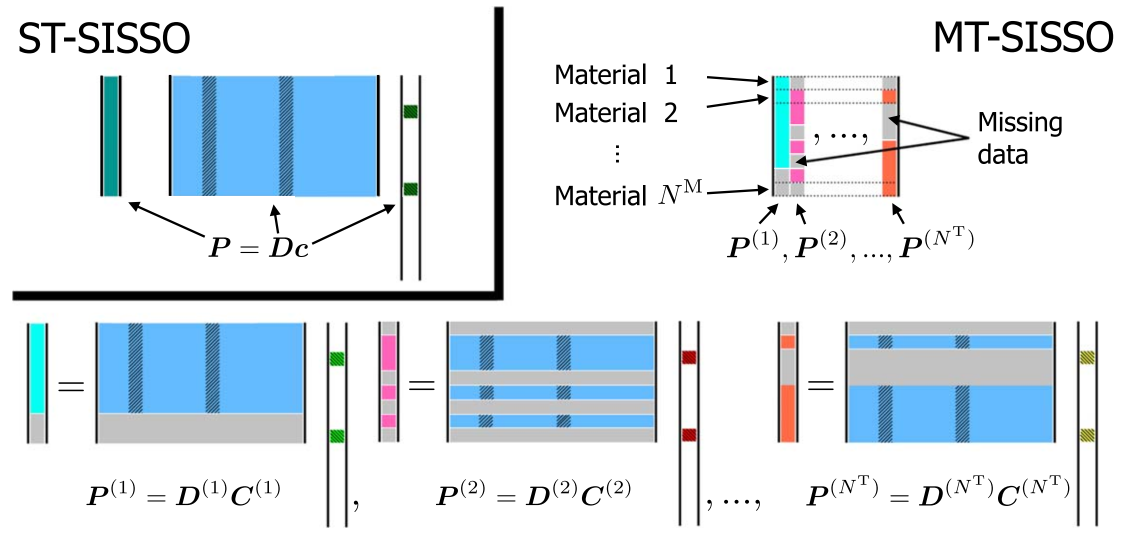
ST-SISSO
$P = Dc$
MT-SISSO
Material 1
Material 2
⋮
Material N^{M}
, …,
Missing data
$P^{(1)}, P^{(2)}, \ldots, P^{(N^{\mathrm{T}})}$
=
$P^{(1)} = D^{(1)}C^{(1)}$
$P^{(2)} = D^{(2)}C^{(2)}$
, …,
$P^{(N^{\mathrm{T}})} = D^{(N^{\mathrm{T}})}C^{(N^{\mathrm{T}})}$

Figure 1. Schematic representation of single-task SISSO versus multi-task SISSO, see section Methodology for reference. The white regions in the (sparse) fitting vectors $\boldsymbol{c}$ and $\boldsymbol{C}^k$ represent zero-valued elements, while the colored-hatched squares represent the nonzero-valued elements, which 'select' the hatched columns in the sensing matrices $\boldsymbol{D}$ and $\boldsymbol{D}^k$. The gray areas in the property vectors $\boldsymbol{P}$ and $\boldsymbol{P}^k$—and correspondingly in the sensing matrices—represent missing/unknown data in the training database, i.e. for some materials (each materials occupies always the same position in the Property vector and is related to the same row in the sensing matrix) some properties are not known. Crucially, the nonzero values of the fitting vectors $\boldsymbol{C}^k$ are always in the same position (they have the same index) for the same MT-SISSO learning, and correspondingly the selected columns are the same in all sensing matrices $\boldsymbol{D}^k$.

difficult to be found. This is the setup of our second showcase application (see section 3.2) where the challenge is to find a model for predicting whether a material is a metal or nonmetal, with materials belonging to many different crystal prototype classes. More specifically, we address the construction of two-dimensional maps where materials being metals or nonmetals are located in two non-overlapping convex regions. In MTL language, each map—one for each crystal prototype—is a *task* and the joint learning imposes that all maps share the same descriptor (in practice the same quantities on the axes). The metal/nonmetal classification challenge was already tackled with (single-task (ST)) SISSO in [25], but here, with an enlarged, heterogeneous materials space (more crystal prototypes), only MT-SISSO is able to achieve an accurate description. Similarly to the previous example, one key feature of the use of MT-SISSO is the possibility to learn predictive models by omitting a significant amount of data from the training database.

Another possible setup of MT-SISSO would be to learn materials properties calculated at different levels of accuracy (e.g. exchange-correlation approximations, but also numerical settings such as basis set type and size, *k*-grid, etc) by imposing the selection of the same descriptor. Also in this case, the advantage would be the possibility to learn from incomplete data, i.e. the training properties are not know at all accuracy levels for all materials. This application would be in the direction of the multi-fidelity learning approach introduced in [40] and will be explored in the future.

Before describing our showcase examples, in the following section we introduce the methodology and notation of MT-SISSO.

2. Methodology

2.1. ST-SISSO for continuous property

In order to underline the analogies and crucial differences between ST and MT-SISSO, we start with a brief recapitulation of the ST-SISSO algorithm. A detailed explanation of the SISSO algorithm is given in [25] and a recommended hands-on tutorial is given in the online Python notebook [41] at the *NOMAD Analytics Toolkit* [42] website. The setup of ST-SISSO starts from a given set of materials with scalar-valued, continuous properties listed in a vector $\boldsymbol{P}$ (an element P_i of $\boldsymbol{P}$ is the property of the *i*th material) and a—typically huge—list of N^{D} possible candidate features forming the *features space*. The projection of each *i*-material into the *j*-feature yields the *i*, *j* component of the 'sensing matrix' $\boldsymbol{D}$, having N^{M} rows and N^{D} columns, with $N^{\mathrm{D}} \gg N^{\mathrm{M}}$. The solution of

$$\underset{\boldsymbol{c}}{\arg\min}\,(\|\boldsymbol{P} - \boldsymbol{D}\boldsymbol{c}\|_2^2 + \lambda\,\|\boldsymbol{c}\|_0), \tag{1}$$

where $\|\boldsymbol{c}\|_0$ is the ℓ_0 norm of $\boldsymbol{c}$, i.e. the number of nonzero components of $\boldsymbol{c}$, gives the optimum Ω-dimensional descriptor, i.e. the set of features singled out by the Ω nonzero components of the solution vector $\boldsymbol{c}$. The parameter λ weights the relative importance of training accuracy versus dimensionality Ω (known as 'sparsity' in the CS language).

The feature space $\boldsymbol{\Phi}_q$ is constructed by starting from a set of primary features $\boldsymbol{\Phi}_0$ and a set of unary and binary operators (such as $+, -$, exp, $\sqrt{\ }$, …). The features are then iteratively combined with the operators, where at each iteration each feature (pair of features) is exhaustively combined with each unary (binary) operator, with the constraint that sums and differences are taken only among homogeneous quantities. The index q in $\boldsymbol{\Phi}_q$ counts how many such iterations were performed. The primary features are typically physical properties of gas-phase atoms (e.g. ionization potential (IP), radius of s ot p valence orbital, etc) and *collective* properties of group of atoms (e.g. formation energy of dimers, volume of the unit cell in a given crystal structure, average coordination, etc) [25]. The features in $\boldsymbol{\Phi}_q$ are represented in terms of mathematical expressions. The evaluation of the jth feature for all the N^{M} materials provides the jth column in the sensing matrix $\boldsymbol{D}$. The properties of gas-phase atoms—in short, *atomic properties*—are 'repurposable', in the sense that they can be used for many descriptor and model learning procedures. For easier reference and reusability, the atomic features used in this work and other related works [20, 24, 25, 30] can be accessed on line at the *NOMAD Analytics Toolkit*. A tutorial [43] shows how to access these quantities and use them in a python notebook.

The algorithm for addressing equation (1) with ST-SISSO is:

(i) SIS preliminary step. A subspace $\boldsymbol{S}_1$ is selected containing the N_1^S features having the largest linear correlation (largest absolute value of scalar product) with $\boldsymbol{P}$. The feature vector $\boldsymbol{d}_1$—the column of $\boldsymbol{D}$ with the largest correlation with $\boldsymbol{P}$—is the one-dimensional ($\Omega = 1$) SISSO solution and also the exact 1D solution of equation (1).

(ii) Evaluation of the residual $\boldsymbol{\Delta}_1 \equiv \boldsymbol{P} - \boldsymbol{d}_1 c_1$, where the scalar $c_1 = (\boldsymbol{d}_1{}^T\boldsymbol{d}_1)^{-1}\boldsymbol{d}_1{}^T\boldsymbol{P}$ is the least-square solution of fitting $\boldsymbol{d}_1$ to $\boldsymbol{P}$.

(iii) SIS step of iteration $\Omega > 1$, which consists in selecting the subspace of the N_Ω^S features with largest correlation with $\boldsymbol{\Delta}_{(\Omega-1)}$ and take the union of this subsets with $\boldsymbol{S}_{(\Omega-1)}$ to form $\boldsymbol{S}_\Omega$.

(iv) SO step of iteration $\Omega > 1$. Several SO strategies are possible; in this paper (as in [25]), we adopt the so-called ℓ_0 regularization, which finds the exact optimum solution within the subset $\boldsymbol{S}_\Omega$ selected by SIS. For all possible Ω-tuples in $\boldsymbol{S}_\Omega$, it finds the one that gives the smallest ℓ_2 (Euclidean) norm of the residual $\boldsymbol{\Delta}_\Omega \equiv \boldsymbol{P} - \boldsymbol{d}_\Omega \boldsymbol{c}_\Omega$, where $\boldsymbol{d}_\Omega$ is the matrix whose columns are the members of the considered Ω-tuple and the vector $\boldsymbol{c}_\Omega = (\boldsymbol{d}_\Omega{}^T\boldsymbol{d}_\Omega)^{-1}\boldsymbol{d}_\Omega{}^T\boldsymbol{P}$ is the least-square solution of fitting $\boldsymbol{d}_\Omega$ to $\boldsymbol{P}$. Points (iii) and (iv) are iterated until the stopping criterion is met. For instance, one stopping criterion (used in the application described in section section 3.1) is that the ℓ_2 norm of $\boldsymbol{\Delta}_\Omega$ is smaller than a prefixed threshold. The Ω-dimensional descriptor identified by ST-SISSO is $\boldsymbol{d}_\Omega$ and the related predictive *model* is $P = \boldsymbol{d}_\Omega \boldsymbol{c}_\Omega$.

The number of iterations q in the construction of the feature space $\boldsymbol{\Phi}_q$ and the dimensionality Ω of the descriptor are (hyper-)parameters of the SISSO method, to be optimized with respect to the validation error of the SISSO model, typically via a class of algorithms known collectively as cross validation (CV). See [25] for the CV strategy for ST-SISSO, while in section 3.1, we discuss CV for MT-SISSO. The size of the subspace selected by SIS, N_Ω^S is also a parameter, but not a hyperparameter to be optimized. In facts, ideally it has to be large enough to include in the set $\boldsymbol{S}_\Omega$ the optimal Ω-dimensional solution contained in $\boldsymbol{\Phi}_q$. In practice, we invoke the relationship that the CS theory establish between size of the feature space, dimensionality of the solution, and number of data points: $N_\Omega^S = \exp\left(N^{\mathrm{M}}/(\kappa \cdot \Omega)\right)$, where κ is a dimensionless constant that the CS theory locates between 1 and 10. We make the further assumption that the number of features added to $\boldsymbol{S}_\Omega$ are the same at each iteration, i.e. N_Ω^S/Ω.

2.2. Multi-task SISSO for learning continuous properties

We denote ($\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \ldots, \boldsymbol{P}^{N^{\mathrm{T}}}$) as the set of N^{T} target property vectors, where each $\boldsymbol{P}^k$ may have a different number of samples, labeled N_k^{M}. $\boldsymbol{D}^k$ is the sensing matrix, with N_k^{M} rows and N^{D} columns, corresponding to the property k. Crucially, all the $\boldsymbol{D}^k$ have the same N^{D}, but possibly different N_k^{M} for different properties $\boldsymbol{P}^k$, $k = 1, 2, \ldots, N^{\mathrm{T}}$. The evaluation of the feature importance for multiple properties needs to consider the overall correlation between a feature and all the properties.

In analogy with ST-SISSO, the MT-SISSO descriptor and model is found by the regularized minimization:

$$\underset{\boldsymbol{C}}{\arg\min} \sum_{k=1}^{N^{\mathrm{T}}} \frac{1}{N_k^{\mathrm{M}}} \|\boldsymbol{P}^k - \boldsymbol{D}^k\boldsymbol{C}^k\|_2^2 + \lambda \|\boldsymbol{C}\|_0, \tag{2}$$

where $\boldsymbol{C}$ is the coefficient matrix, with N^{D} rows and N^{T} columns, i.e. its kth column $\boldsymbol{C}^k$ is the vector of coefficients fitting $\boldsymbol{D}^k$ to $\boldsymbol{P}^k$. The ℓ_0 norm of the matrix $\boldsymbol{C}$ counts the number of *rows* that have at least one nonzero element. In practice, for each property a separate least-square regression is performed and what is minimized is the average squared error over all the regressions. The regularization imposes that when a feature $\boldsymbol{D}_j^*$ (the set of columns j of all the $\boldsymbol{D}^k$) is selected (i.e. it has nonzero coefficient C_j^k) for one property k, then it is selected for all properties. Mathematically, this regularization across properties (tasks) stabilizes the descriptor selection also with data unevenly distributed over the different properties. The model for any property k is $\boldsymbol{P}^k = \boldsymbol{D}^k\boldsymbol{C}^k$, where each $\boldsymbol{C}^k$ has the nonzero elements at the same indexes $\{j_1, j_2, \ldots, j_{N^{\mathrm{D}}}\}$, i.e. the same features are selected for all properties. From a physical point of view, it is desirable that the different properties are homogeneous so that it *makes sense* that the same descriptor maps into all properties, albeit with the crucial flexibility of different fitting coefficients.

Similarly to ST-SISSO, the MT-SISSO solution of equation (2) starts with a SIS step. To extend the SIS scheme for feature ranking with multiple properties, we first standardize all the features, i.e. the average $\overline{D_j^k}$ over all samples N_k^{M} is subtracted from each feature column vector $\boldsymbol{D}_j^k$ and the result is divided by its standard deviation: $\boldsymbol{D}_j^k \to (\boldsymbol{D}_j^k - \overline{D_j^k}) \big/ \|\boldsymbol{D}_j^k - \overline{D_j^k}\|_2$. In this way, the absolute values of the linear correlations (scalar product) of every feature with a given property P^k are comparable. We note that the standardization is the final operation *after* the matrices $\boldsymbol{D}^k$ are constructed following the iterative procedure described above for ST-SISSO. When the features are combined with the operators, their values are not yet standardized.

In the first iteration of the MT-SISSO algorithm, we have only a SIS step: the overall correlation of a feature j (the jth column of the sensing matrix D_k for the kth property) with all the properties is defined as quadratic mean of their scalar products:

$$\theta_j = \sqrt{\sum_{k=1}^{N^{\mathrm{T}}} < \boldsymbol{D}_j^k, \boldsymbol{P}^k >^2 / N^{\mathrm{T}}}. \tag{3}$$

SIS ranks the features according to θ_j and collects in $\boldsymbol{S}_1$ the top N_1^S features to form a subspace. Also for MT-SISSO, the feature with highest θ_j is already the optimum 1D descriptor.

Next, the set of residuals $(\boldsymbol{\Delta}_1^{(1)}, \boldsymbol{\Delta}_1^{(2)}, \ldots, \boldsymbol{\Delta}_1^{N^{\mathrm{T}}})$ is evaluated, using $\boldsymbol{\Delta}_1^k \equiv \boldsymbol{P}^k - \boldsymbol{d}_1^k\boldsymbol{c}_1^k$, analogous to the ST-SISSO approach discussed above.

At the second and each subsequent iteration of MT-SISSO we have a SIS and a SO step. In the SIS step at iteration $\Omega > 1$, θ_j is evaluated as in equation (3), with $\boldsymbol{\Delta}_{(\Omega-1)}^k$ instead of P^k, and the newly selected subset of features is added to $\boldsymbol{S}_{(\Omega-1)}$ to form $\boldsymbol{S}_\Omega$.

In the SO step at iteration $\Omega > 1$, all possible Ω-tuples in $\boldsymbol{S}_\Omega$ are formed. If $\boldsymbol{d}_\Omega^*$ is the matrix whose columns are the members of one considered Ω-tuple, $\boldsymbol{d}_\Omega^k$ its sub-matrix with entries related to the samples with properties $\boldsymbol{P}^k$, and $\boldsymbol{c}_\Omega^k = (\boldsymbol{d}_\Omega^{kT}\boldsymbol{d}_\Omega^k)^{-1}\boldsymbol{d}_\Omega^{kT}\boldsymbol{P}^k$ is the least-square fit of $\boldsymbol{d}_\Omega^k$ to $\boldsymbol{P}^k$, then the Ω-tuple that minimizes $\sqrt{(\sum_{k=1}^{N^{\mathrm{T}}}\frac{1}{N_k^{\mathrm{M}}}\|\boldsymbol{P}^k - \boldsymbol{d}_\Omega^k\boldsymbol{c}_\Omega^k\|_2^2)/N^{\mathrm{T}}}$ is the identified Ω-dimensional descriptor.

2.3. MT-SISSO for categorical properties

Besides continuous properties, materials can be classified by means of categorical properties (e.g. being metal, nonmetal, topological insulator, etc) into classes. In this work, we present MT-SISSO for classification in the following way: we consider as one *task* the construction of one materials-property map (with two or more classes, i.e. values of the considered categorical property). A map is a low-dimensional representation of the materials space where each material is located by means of an appropriate descriptor vector (the components of the descriptor are the coordinates in the low-dimensional representation) such that all materials sharing a certain categorical property are located in the same convex region. In a good/useful map, regions containing materials with exclusive properties (e.g. metals versus nonmetal) do not overlap. In a general materials-property map, the regions assigned to a certain class do not need to be in a convex region, actually not even in a connected region. However, in order to design a computationally efficient algorithm, we impose that the regions are convex, with some loss of generality.

The MT-SISSO formulation of the classification problem is to find multiple maps for subsets of materials that share a common descriptor, but possibly differently positioned boundaries between classes. The materials are grouped into subsets by categorical physical properties, such as bonding type, space group, etc. As introduced in [25], the mathematical formulation of ST-SISSO for classification adopts a measure of the overlap between convex regions as quantity to be minimized by the optimization algorithm. For a property with N^{C}

classes [25]:

$$\underset{\boldsymbol{c}}{\arg\min} \sum_{I=1}^{N^{\mathrm{C}}-1} \sum_{J=I+1}^{N^{\mathrm{C}}} O_{IJ}(\boldsymbol{D}, \boldsymbol{c}) + \lambda \|c\|_0, \tag{4}$$

where $O_{IJ}(\boldsymbol{D}, \boldsymbol{c})$ is the number of data in the overlap region between the I-domain and these J-domain, $\boldsymbol{c}$ is a vector with elements 0 or 1, so that a feature k (the kth column of $\boldsymbol{D}$ is selected (deselected) when $c_k = 1(0)$), and λ is a parameter controlling the number of nonzero elements in $\boldsymbol{c}$. O_{IJ} depends on $(\boldsymbol{D}, \boldsymbol{c})$ in the sense that the nonzero values of $\boldsymbol{c}$ select features from $\boldsymbol{D}$ that determine the position (coordinates) of the data and the shape of the convex region in the map. The MT-SISSO classification formulation for 'multi-map' learning is simply:

$$\underset{\boldsymbol{C}}{\arg\min} \sum_{k=1}^{N^{\mathrm{T}}} \sum_{I=1}^{N^{\mathrm{C}}-1} \sum_{J=I+1}^{N^{\mathrm{C}}} O_{IJ}(\boldsymbol{D}^k, \boldsymbol{C}^k) + \lambda \|\boldsymbol{C}\|_0, \tag{5}$$

where a feature (a column of $\boldsymbol{D}^k$) is selected for all maps, or none, and the index k runs over the tasks, i.e. the maps.

The MT-SISSO solution of equation (5) involves a SIS and a SO step. In the SIS step, the following expression is evaluated:

$$\theta_j = \left(\sum_{k=1}^{N^{\mathrm{T}}} \sum_{I=1}^{N^{\mathrm{C}}-1} \sum_{J=I+1}^{N^{\mathrm{C}}} O_{IJ}^{\mathrm{1D}}(\boldsymbol{d}_j^k) + 1 \right)^{-1}, \tag{6}$$

where $O_{IJ}^{\mathrm{1D}}(\boldsymbol{d}_j^k)$ is the number of points in the overlap *interval* between the I-domain and these J-domain when all data points (related to property k) are represented via the (one-dimensional, 1D) descriptor $\boldsymbol{d}_j^k$ (i.e. the jth column of $\boldsymbol{D}^k$). In other words, all materials are projected onto a 1D coordinate, defined by each of the columns of the sensing matrix. Thinking for simplicity at only two classes A and B, O_{AB}^{1D} counts how many points (if any) are in the overlap interval between the intervals occupied by points in class A and B. The index θ_j has range (0, 1], with large value corresponding to fewer data in the overlap region between domains; $\theta_j = 1$ indicates no overlap between any two domains. Similarly to the continuous-valued property case, the N_1^S features $\boldsymbol{d}_{j_1}^k, \boldsymbol{d}_{j_2}^k, \ldots, \boldsymbol{d}_{N_1^S}^k$, with smallest overlap (largest θ_j) are selected into the subset $\boldsymbol{S}_1$. Here, the 'residual' is the set of data points in the overlap regions. This means that, at any subsequent iteration, SIS looks for the 1D feature that better classifies the data points that are not classified at the previous iterations. The newly selected features are added as usual to $\boldsymbol{S}_{(\Omega-1)}$ in order to build $\boldsymbol{S}_\Omega$.

In the SO step at iteration $\Omega > 1$, all the Ω-tuples in $\boldsymbol{S}_\Omega$ are listed and the Ω-tuple that minimizes $\sum_{k=1}^{N^{\mathrm{T}}}\sum_{I=1}^{N^{\mathrm{C}}-1}\sum_{J=I+1}^{N^{\mathrm{C}}} O_{IJ}(\boldsymbol{d}_\Omega^k l)$ is the selected Ω-dimensional descriptor.

Besides the domain overlap O, other metrics exist for classification, e.g. the number of misclassified data as defined by a support-vector machine (SVM) built with all the Ω-tuples in S_Ω, as adopted in [30].

2.4. Computational complexity of SISSO

The time complexity for the SIS step of the SISSO algorithm is linear with the number of training data N^{M} and the size of feature space N^{D}, i.e. $O(N^{\mathrm{M}} \cdot N^{\mathrm{D}})$, [26]. For the SO step (in the ℓ_0-regularization implementation as discussed in this paper), the time complexity depends on whether the target property is continuous (regression problem) or categorical (classification problem). Though the ℓ_0 regularization is formally NP hard, it can be made feasible by restricting to low dimension of the descriptor and moderate size of features subspace selected by SIS. With the total SIS-selected subspace size N_Ω^S and the descriptor dimension Ω, the time complexity of SO with ℓ_0 for continuous property is $O\left(N^{\mathrm{M}} \cdot (N^{\mathrm{D}})^2 \cdot \binom{N_\Omega^S}{\Omega}\right)$, where $N^{\mathrm{M}} \cdot (N^{\mathrm{D}})^2$ is the time needed for evaluating one candidate model using least-square regression and the binominal coefficient $\binom{N_\Omega^S}{\Omega}$ is the total number of candidate models to be evaluated. For classification problems targeting two-dimensional maps, the time scaling of SO with ℓ_0 is $O\left((N^{\mathrm{M}})^2 \cdot \binom{N_\Omega^S}{\Omega}\right)$, where $(N^{\mathrm{M}})^2$ is the time needed for evaluating one candidate model.

3. Results and discussion

3.1. MT-SISSO for the relative stability of different structure pairs of *AB* binary materials

In [20, 24, 25] the learning of the relative stability between the RS and ZB structures of *AB* octet-binary compounds was used as showcase study. Here, we address, again for the octet binaries, the relative stability of five

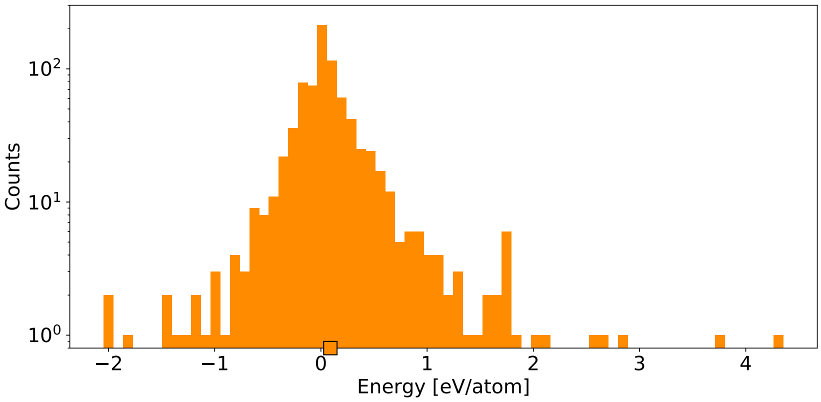

Figure 2. Distribution of reference (DFT-LSDA) energy differences (10 pairs of structures) for all 82 octet binaries. The square marks the average value of the distribution.

crystal structures, including RS and ZB and we add add three more crystal structures: the CsCl, NiAs, and CrB prototypes. The prediction of relative stability among several structures is naturally suited for MTL and in particular MT-SISSO.

As a dataset, we used the same 82 octet binaries as in [20, 24, 25], although now each of the binary material was optimized in five different crystal structure prototypes, by fully relaxing all degrees of freedom compatible with the crystal symmetry (1 degree of freedom for RS, ZB, and CsCl, 2 degrees of freedom for NiAs, and 5 for CrB). Forces and energies were evaluated via DFT using the local-spin-density approximation (LSDA). The calculations were performed with FHI-aims [44] using the high precision third-tier basis set with 'tight settings' for the numerical integration grids. The total energies of the data are estimated to be converged below 10 meV/atom and the energy differences between structures below 5 meV/atom. More information on these high-throughput DFT calculations can be found in [45] and all inputs and outputs are in the NOMAD repository.

For the descriptor identification, we use atomic properties as input features: the IP, electron affinity (EA), number of valence electrons n_{val}, the group number G in the periodic table, and the radii $r_{s,p,d}$ where the radial probability density of the valence s, p, and d orbitals are maximal. Furthermore, equilibrium distances d_{ij} of homonuclear AA and BB, and AB dimers are included. All the features were calculated with the LSDA. In the *NOMAD Analytics Toolkit*, also other sets of atomic features, calculated with other exchange-correlation functionals, are provided. Our experience is that the set of features used to build $\mathbf{\Phi}_0$ should be consistent, i.e. calculated with the same model Hamiltonian or measured with the same methodology. It is not necessarily true, however, that target properties and features in $\mathbf{\Phi}_0$ should be consistent. For instance, one may predict experimentally measured quantities starting from DFT features.

We set the parameter κ that determines the sizes of the SIS subspaces to 3.3. With $N^{\mathrm{M}} = 82$, the subspace sizes N_{Ω}^{S} are approximately $2 \times 10^5, 4 \times 10^3, 5 \times 10^2$, and 10^2 for $\Omega = 2$–5. These values are kept fixed through all our numerical test, e.g. also when N^{M} is decreased in the CV tests. For the routine application of ST and MT-SISSO, we note that the sizes N_{Ω}^{S} are rather large for the features space used in this work. We checked that even for $\kappa = 4$, the same descriptors are always found at $\Omega = 2$, while for $\Omega = 3$ even $\kappa = 5$ is small enough to yield the same descriptor as for $\kappa = 3.3$.

Starting from the DFT reference cohesive energy (total DFT energy minus the total DFT energy of the gas-phase ground-state atoms) of the five crystal structures for all the octet-binary materials, we constructed 10 sets of all the possible energy differences between two crystal structures. Each energy difference is then a task in a MT-SISSO learning. In figure 2, we show the distribution of these energy differences.

The main purpose of this showcase application is to learn a phase diagram (a map) where different non-overlapping regions of the diagram contain the materials with the same ground-state structure. This is similar conceptually to the classification-driven construction of materials-property maps discussed in the next section, but the crucial difference is that we target a continuous property (energy) and only *a posteriori* we determine the most stable phase (i.e. the ground-state crystal structure) for each material, simply by identifying which phase is predicted to have the lowest-energy for each material. We emphasize that higher-energy (meta-stable) structures are learned as well. The fact that predicting energies leads to phase diagram is embedded in the fact that the MT-SISSO models are linear with the descriptor (which determines the coordinate of each material in the map), found by the MT-SISSO algorithm. With the purpose of the phase diagram creation in mind, it should become evident why, physically, MTL is the obvious framework to use. Having one descriptor for all target properties allows to represent all the (linear) models with the same axes, resembling a traditional phase diagram with the component of the descriptor found by SISSO acting as the familiar order/control parameters.

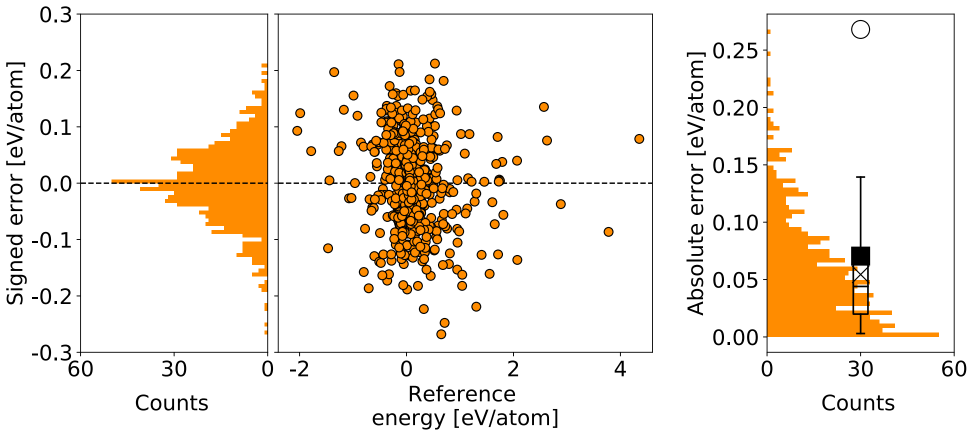

Figure 3. Central panel: training errors versus reference energies of the MT-SISSO fits to the energy differences. Left panel: distribution training errors for the same fit, obtained by integrating the central-panel plot over the reference energies. Right panel: distribution of *absolute* training errors and corresponding 'box plot'. The box plot marks the 25th and 75th percentiles (extrema of the rectangle), the 5th and 95th percentiles (extrema of the 'whiskers'), and the median (horizontal line inside the rectangle). Also shown are the mean absolute error (MAE, cross), the root mean square error (RMSE, solid square), and the maximum absolute error (MaxAE, circle). The feature space Φ_3 and dimension $\Omega = 3$ were used.

The choice of having all the energy differences as tasks is important in order to build a phase diagram for the phase (crystal structure) stability, when using a linear MTL like MT-SISSO. While only four energy differences (for five crystal structures) are independent, the simultaneous learning of all energy differences limits the prediction error of the relative stability between all phases. In contrast, using only one structure as reference and learning the energy difference from that structure may lead to large errors for the relative stability of any two other phases. Furthermore, a subtle implication of the MT-SISSO learning of all possible energy differences is that the models maintain an *internal consistency* with respect to a common energy zero. In practice, for any three structures α, β, γ, the difference in energy $E(\alpha) - E(\gamma)$ is by construction equal to $(E(\alpha) - E(\beta)) - (E(\gamma) - E(\beta))$. This is not (necessarily) true if the three energy differences are learned with separate, independent models. We will come back to this aspect when discussing the phase diagram derived from the learned MT-SISSO models.

In figure 3, we show the training errors of the MT-SISSO model for the energy differences, trained by using the feature space $\mathbf{\Phi}_3$ and dimensionality $\Omega = 3$ (see further for the justification of this choice). The overall root mean square errors (RMSE) errors, 0.07 eV/atom, should be compared to the standard deviation of the reference data distribution, which is 0.49 eV/atom. The latter value represents the so-called *baseline*, i.e. the RMSE for the model that predicts for all points the average values of the target property over the training data.

In order to give a feeling on the computational effort necessary to find the MT-SISSO descriptor and model, we report that the learning for the settings described above was run on an Intel Xeon E5-2698 v3 node with 2 CPUs per node (16 cores/CPU @ 2.3 GHz) and it took 5 h. On a 4-cores laptop, it would take a couple of days of runtime. We remind that the size of the features space is huge: 2×10^{10} features.

Here, we note that the MT-SISSO approach can be also seen as a way to include collective or structural features of the materials, such as the local environment of each atom, in the learning scheme. Rather than trying to explicitly include a functional dependence of the local environments, the different environments (here, the different crystal structure prototypes) are assigned to different tasks and to each local environment is assigned a different set of coefficients for the mapping of the common (environment-independent) descriptor found by MT-SISSO.

In figure 4 (the corresponding numerical values are tabulated in table 1), we show the CV test for the energy difference learning, performed in order to assess the two hyperparameters of MT-SISSO: the (size of the) feature space $\mathbf{\Phi}_q$ and the dimensionality Ω of the descriptor. To the purpose, we performed a leave-10%-out CV, i.e. 10% of the materials are left out of the training set, the MT-SISSO model is trained on the remaining 90% of the materials, and the errors are measured for the left-out materials. This random selection of training and validation sets was repeated 30 times, which we found sufficient to converge the validation RMSE to 0.01 eV. We note (a) that all the 10 target properties of a material are excluded from the training set when it is left out and (b) the standardization of the features is performed at each random selection of the training set, only on the features relative to the actual training data points. This latter highly recommended practice is crucial to avoid information 'contamination' between the training and validation set.

Analysis of figure 4 reveals that models trained by using the larger feature space $\mathbf{\Phi}_3$ (containing $\sim 2 \times 10^{10}$ features) are consistently better performing (in terms of prediction errors) than models trained starting form $\mathbf{\Phi}_2$

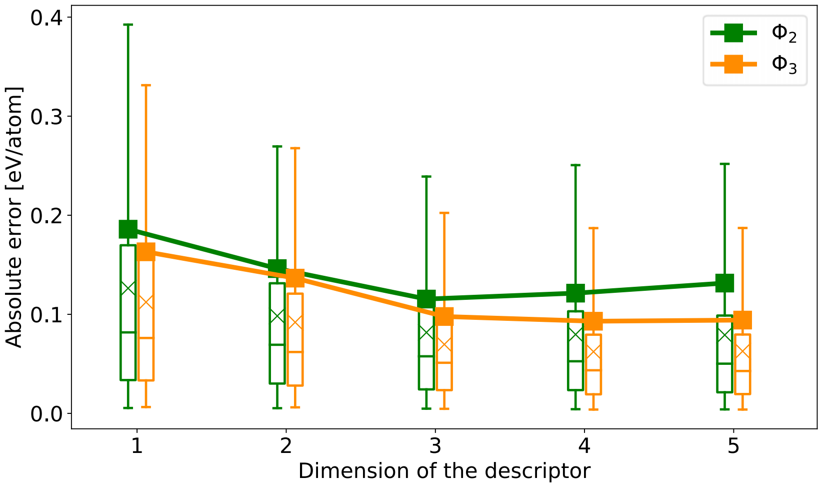

Figure 4. Cross validation prediction errors of MT-SISSO models for the energy difference learning, as function of the dimension of the descriptor Ω, for the feature space Φ_2 and Φ_3. All errors are averaged over 30 repetitions of leave-10%-out CV (MT-SISSO is trained over 90% of randomly selected data and tested on the remaining 10%). The box plots and symbols are consistent with figure 3, right panel.

Table 1. Tabulated values from figure 4. p_{75} and p_{95} are the 75th and 95th percentiles, respectively, RMSE is the root mean square error, and MaxAE is the maximum absolute error. All quantities are given in (eV/atom).

	Ω	RMSE	Median	p_{75}	p_{95}	MaxAE
$\boldsymbol{\Phi}_2$	1	0.186	0.082	0.170	0.393	1.098
	2	0.146	0.069	0.131	0.272	1.055
	3	0.115	0.058	0.112	0.240	0.649
	4	0.121	0.053	0.103	0.252	0.968
	5	0.132	0.050	0.099	0.252	1.385
$\boldsymbol{\Phi}_3$	1	0.163	0.076	0.158	0.332	1.056
	2	0.137	0.062	0.121	0.268	0.973
	3	0.098	0.051	0.090	0.205	0.548
	4	0.093	0.043	0.079	0.187	0.742
	5	0.094	0.043	0.080	0.189	0.709

(containing ~2.4×10^5 features), for all dimensions. RMSE and mean absolute errors are only marginally better when going from $\boldsymbol{\Phi}_2$ to $\boldsymbol{\Phi}_3$, but we notice that the largest percentiles (75th and 95th) improve significantly, especially for $3 \leqslant \Omega \leqslant 5$. Looking at larger percentiles of the error distributions, besides looking at mean errors, is important because, for a predictive model, we are typically interested that the worst cases still yield relatively small errors. The overall best model is ($\boldsymbol{\Phi}_3$, $\Omega = 5$), but we also notice that, for $\boldsymbol{\Phi}_3$, the improvement of all error indicators when going from $\Omega = 3$ to $\Omega = 5$ is only marginal. Therefore, in view of the significantly smaller computational time needed to train $\Omega = 3$ versus $\Omega = 5$, in the following tests, we focus on ($\boldsymbol{\Phi}_3$, $\Omega = 3$), starting from figure 5, where we report the detailed analysis of the signed and absolute errors for these latter settings.

We now turn our attention to two tests that reveal the peculiarity of MTL versus traditional ST learning when only incomplete data are available. In the first test, we selected left-out sets in this way: one material and one crystal structure are randomly selected and all the energy differences involving the selected structure are eliminated from the training set for the selected material. The procedure is repeated until a prefixed x% of pairs (material, structure) are eliminated (we recall the total number of such pairs is $82 \times 5 = 410$). This test simulates the training over a materials database where for some (or many) materials the information for only some crystal structures is available. It would be of great value if from such dishomogenous database, one could predict the missing information. For a meaningful test, we added the following two constraints in the simulated elimination of database fields: for each material, the energy of at least two crystal structures is known and for each of the ten tasks (energy differences) there are at least four materials carrying the information, in order to have enough data to train the four fitting coefficients of the $\Omega = 3$ model. For each x% selected value, we train one MT-SISSO model and ten independent ST-SISSO models (one for each task of MT-SISSO). We then look at

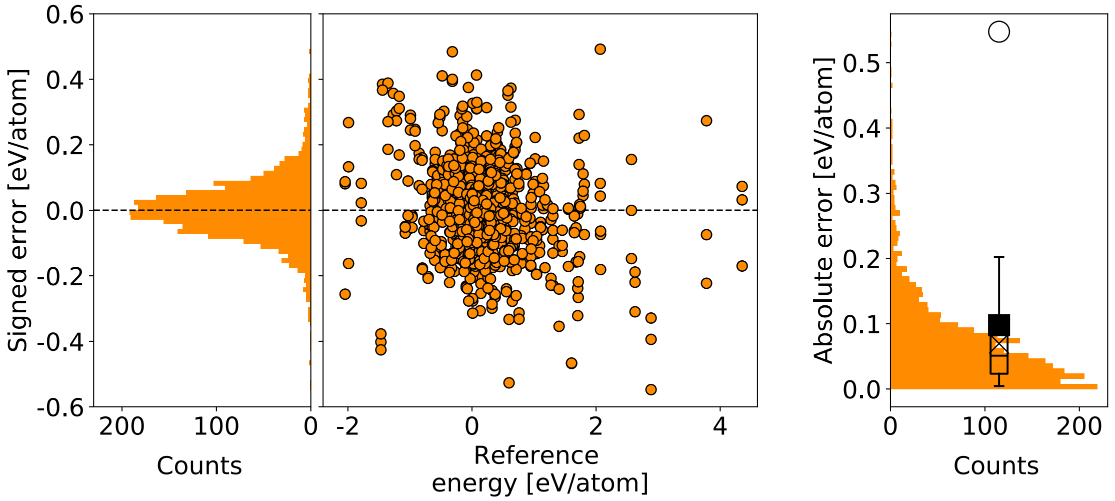

Figure 5. Same set of quantities as in figure 3, for the signed and absolute *prediction* errors for ($\Phi_3, \Omega = 3$).

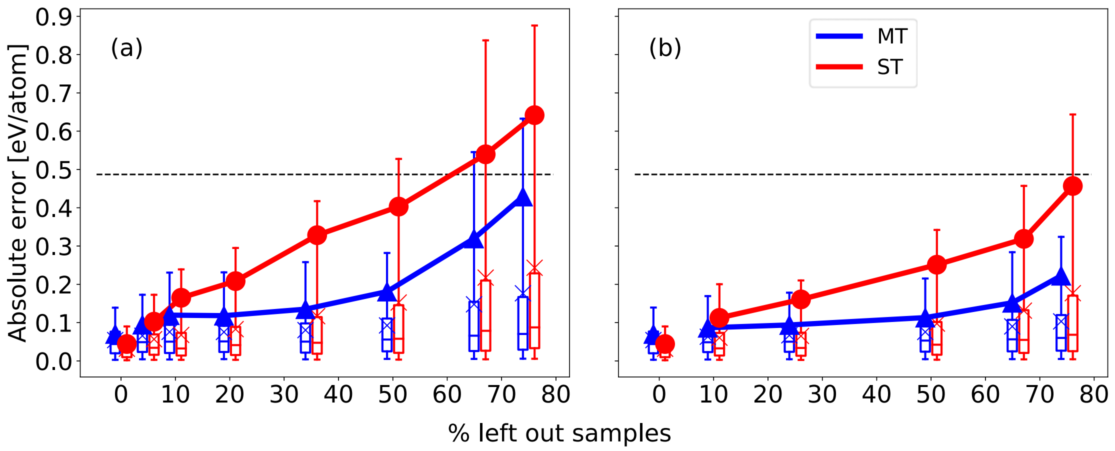

Figure 6. Prediction errors of MT-SISSO versus (average) ST-SISSO for (left panel) 'leave x% of (materials, structure) data out' and (right panel) 'leave y% of data for one crystal structure out'. The symbol convention is the same as in figure 3. In both panels, the errors at 0% data out in both panels are training errors. The symbols at the same % are slightly staggered in order to improve visibility. For reference, the actual values are 0%, 5%, 10%, 20%, 35%, 50%, 66%, 75%. The horizontal line at 0.49 eV is the baseline (see text).

the prediction errors on the missing data. Figure 6(a) shows the outcome of the test. With abuse of notation, the values at 0% refer to training error. As one should expect, ST-SISSO yields lower training error due to higher flexibility (for each task, a different descriptor can be chosen). However, as soon as data are missing, MT-SISSO rules with lower RMSE and, crucially, with lower largest errors. Interestingly, the quality of MT-SISSO stays pretty unchanged, for all error indicators, over a wide range of amount of missing data.

In the second test, we selected one crystal structure and then we removed the energy values for a given y% of materials. Removing the energy value of one structure implies the removal of four energy differences from the (material, energy differences) database. One MT-SISSO model and four ST-SISSO models are trained and the errors for the selected structures are evaluated on the missing materials. This test simulates the case of a new crystal structure being identified for only few materials in the database and one wants to learn with the fewest possible data the predicted energy in such new crystal structure for all materials. Figure 6(b) shows the performance of the MT-SISSO model versus the average of the four ST-SISSO model. Again the training error (at 0%) favors ST-SISSO and again MT-SISSO's predictive performance remain impressively constant over a wide range of amount of missing data.

These two tests show numerically what should be expected from a physical point of view: it is reasonable to assume that the energy of different crystal structures depend on the same mechanism encoded in the properties of the gas-phase atoms used as primary features. Therefore MT-SISSO uses at best the (possibly scarce) information scattered over all crystal structures to identify such mechanism. In this way the prediction on the

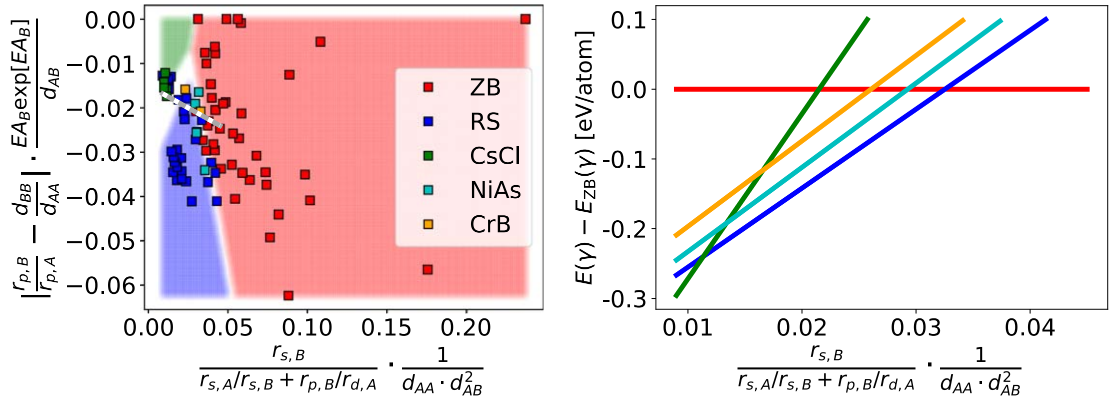

Figure 7. Left: MT-SISSO-learned phase diagram (structure map) of the ground-state crystal structure for the octet binaries. The colored areas represent the predicted stability region for the structure with the same color in the legend. The squares are colored according to the reference lowest-energy structure. The white color marks areas where the difference between the energy of the lowest-energy and the second lowest-energy structures differ by at most 0.03 eV/atom. Right: cut of the phase diagram along the dashed line shown in the left panel. The lines are the traces of the planes representing the predicted energy difference from the baseline (ZB structure).

scarcely known materials and/or crystal structures is more reliable than a model that uses information from only one crystal structure (or, one pair of crystal structures, as in the presented case) to identify the descriptor.

We close the section on MT-SISSO by showing how the ($\Omega = 2$) MT-SISSO model trained over all data points can be used to draw a phase diagram (crystal structure map). The model identified by MT-SISSO for each task can be represented as a plane in a 3D space, where the coordinates (x, y) are the components of the descriptor and coordinate z is the predicted energy. The mentioned property of *internal consistency* among MT-SISSO models for (energy) differences allows for the unambiguous determination of the predicted lowest-energy structure for each coordinate (x, y). A color is associated with any specific crystal structure and assigned to a square (pixel) $(\delta x, \delta y)$ centered on (x, y) when the corresponding structure is the lowest in energy at (x, y). Figure 7(a) represents the structure map for the octet binaries. The colored area refer to the predictions and the colored squares are the reference data. The white color marks areas where the energy difference between the lowest-energy and the second lowest-energy structures differs by less than 0.03 eV/atom. In order to give an insight into the 3D visualization of the structure map, we show in figure 7(b), a cut along the gray-white dotted line marked in figure 7(a). This shows that some crystal structures are predicted to be very close in energy for certain values of the descriptors. In a realistic application, one may conclude that the actual ground-state in the neighborhood of those values of the descriptor may be any of the low-energy structures (in particular, at finite temperature), while those that are predicted to be very high in energy can be safely discarded as candidate ground-state. To gauge the trustfulness of the presented phase diagram, we mention that the largest prediction error for a structure that appears 'misclassified' (the color of its symbol does not match the background—predicted—color) is 0.09 eV/atom.

3.2. MT-SISSO for the metal/insulator classification of $A_x B_y$ binary materials

In [25], a SISSO-trained model for the metal/insulator classification of 299 binary materials distributed over 15 prototypes was presented, with (experimental) reference data collected from the SpringerMaterials database [46]. That model achieved 99% classification accuracy with a 2D descriptor, but had several constraints, i.e. ignoring materials of certain bonding types. In the present work, we extend the metal/insulator dataset to totally 334 A_xB_y binary materials (197 metals and 137 nonmetals) belonging to 17 crystal structure prototypes. The new dataset includes the 15 three-dimensional prototypes previously considered [25] and, in addition, two layered prototypes: CdI_2 and MoS_2. The pie-chart of the distribution of data points over prototypes is shown in figure 8. The descriptor described in [25] was a function of properties of gas-phase atoms plus one collective feature, namely the unit cell volume. At first, by using the same set of primary features, we check whether SISSO can find a single map that correctly classifies into metal versus nonmetals the materials in all 17 prototypes. Specifically, we considered as primary features: {ionization energy IE, Pauling electronegativity χ, covalent radius r_{cov}, unit cell volume normalized by total atom volume $V_{\text{cell}}/\sum V_{\text{atom}}$, bonding distance in the material between A and B d_{AB}, coordination number of A species N_A^{N} and of B species N_B^{N}, and atomic fraction for A x_A and B x_B}. As in [25], the values for the atomic features are taken from WebElements [47] and the information for building the structural features (atomic coordinates, species, and lattice vectors) comes from the SpringerMaterials [46] database. Furthermore, we considered as operator set: $\{+, -, \times, /, \exp, \log, |-|, \sqrt{\ }, ^{-1}, ^2, ^3\}$. From these ingredients, we build the feature sapce $\mathbf{\Phi}_3$. The size of the SIS-selected subspace for each descriptor dimension

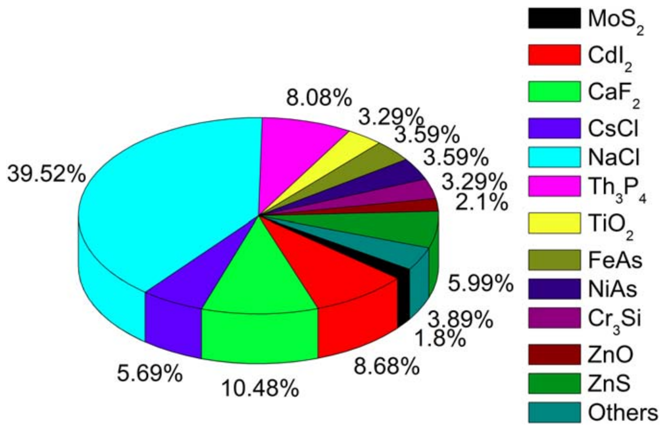

Figure 8. Pie-chart showing the distribution of the 334 reference binary materials, taken from SpringerMaterials, over the 17 considered crystal structure prototypes.

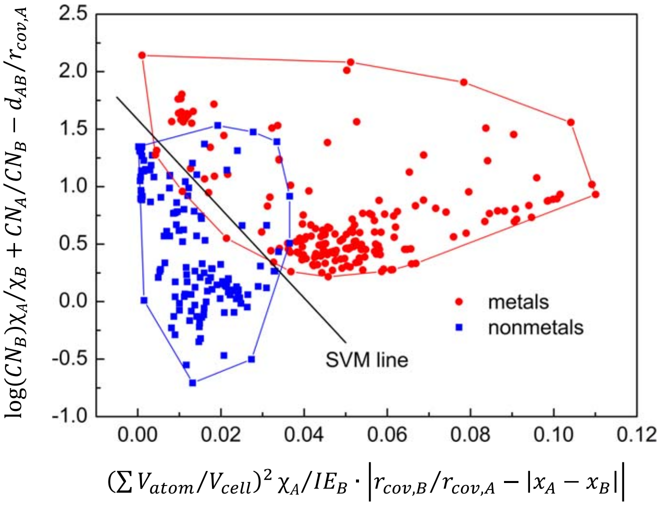

Figure 9. The metal/nonmetal classification map for binaries on all 17 prototypes. The (red-) blue-bordered convex regions denote the (metal) nonmetal domain. The linear-SVM-trained separation line was found with the 2D descriptor fixed to the one found by SISSO.

was set to 10^4 which is a big yet manageable size for descriptors up to 2D. Unless otherwise stated, these settings are used for all the classification problems discussed below. Figure 9 shows the classification map by the best SISSO-trained 2D descriptor. There is an overlap between the metal and nonmetal regions, and in total there are 36 data points in the overlap region. Among the materials in the overlap, 13 (8 metals and 5 nonmetals) are in the CdI$_2$ prototype, and 6 (1 metal and 5 nonmetals) are in the MoS$_2$ prototype. For the latter prototype, we have information only on 6 materials. The other 17 materials in the overlap belong to the other 15 prototypes. In the map of figure 9, the optimal separation line was found by using a linear SVM with the SISSO-determined 2D descriptor. According to the SVM metric, 17 out of 334 materials are misclassified. To avoid confusion, in the following the number of misclassified data points will always refer to the SVM metric, while as SISSO figure of merit we report the 'number of data point in the overlap region'. It is not strictly necessary to apply SVM after SISSO, as SISSO for classification already targets a map that separates as much as possible (ideally, fully, without overlap) the different classes of materials. However, the SISSO model is determined by all the boundary materials defining the convex regions. An SVM line (at fixed descriptor determined by SISSO) is a well defined and a much simpler model, which does not conflict with the SISSO model.

Though a global descriptor (up to 2D) for the accurate metal/insulator classification of all prototypes is not found with the current primary features, the independent classification for each prototype with 100% training accuracy is very easy to achieve. Table 2 shows the simple 1D descriptors for 100% classification of metal/insulator of the binary materials for each prototype independently. Actually, ST-SISSO finds many descriptors for the 100% classification within each prototype, and table 2 shows only the most simple ones (with least

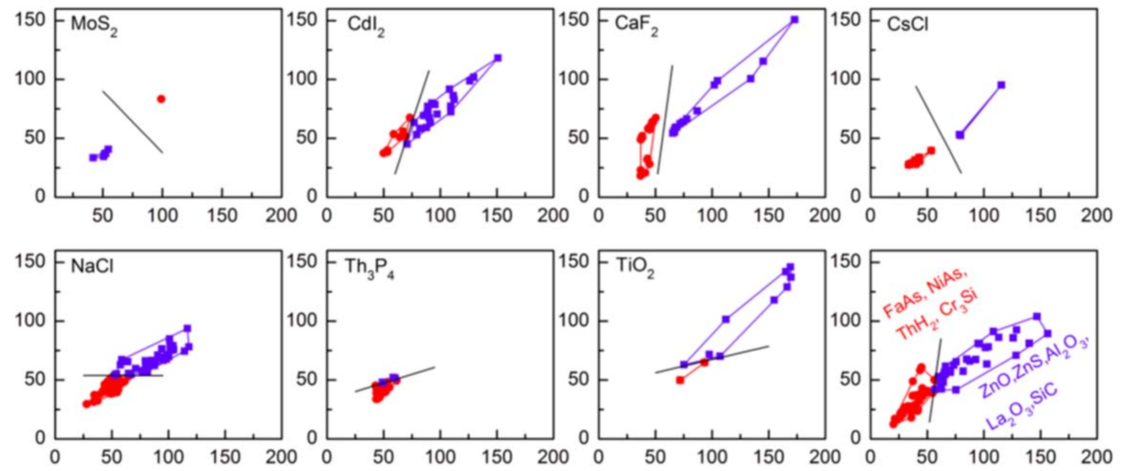

Figure 10. MT-SISSO results for the classification of metal/nonmetal for 17 crystal prototypes. The component of the SISSO-determined descriptor on the x- and y-axes (the same in all plots) are given in equation (7). There is zero overlap between the metal and nonmetal domains on all the maps. The separation lines were found via linear SVM.

Table 2. Descriptors yielding metal/nonmetal 100% classification accuracy within each prototype. The primary features of IE, χ, $V_{\rm cell}/\sum V_{\rm atom}$, and d_{AB} were used for these calculations; coordination number $N^{\rm N}$ and atomic fraction x were excluded because they are constant within one prototype. Since all descriptors are one-dimensional, we also provide the threshold values for the metal/nonmetal transition (metals are for values of the descriptor smaller than the threshold).

Prototype[a]	Number of data	Descriptor	Boundary
MoS$_2$	6 (1 metal, 5 nonmetals)	χ_A	1.68
CdI$_2$	29 (8 metals, 21 nonmetals)	$d_{AB}\chi_B^3$	41.08
CaF$_2$	35 (21 metals, 14 nonetals)	χ_B	2.68
CsCl	19 (16 metals, 3 nonmetals)	IE_B	9.55
NaCl	132 (87 metals, 45 nonmetals)	$\frac{V_{\rm cell}}{\sum V_{\rm atom}}\frac{\mathrm{IE}_A\mathrm{IE}_B r_{{\rm cov}A}}{\chi_A}$	135.79
Th$_3$P$_4$	27 (23 metals, 4 nonmetals)	$\frac{V_{\rm cell}}{\sum V_{\rm atom}}\mathrm{IE}_A(d_{AB}\mathrm{IE}_B)^2$	676.24
TiO$_2$	11 (2 metals, 9 nonmetals)	$-\chi_A$	−2.105
{FeAs, NiAs, ThH2, Cr$_3$Si, ZnO, ZnS, Al$_2$O$_3$, La$_2$O$_3$, SiC}[b]	73 (38 metals, 35 nonmetals)	$\frac{V_{\rm cell}}{\sum V_{\rm atom}}\mathrm{IE}_B\chi_B$	42.90

[a] ReO$_3$ prototype was not considered because of only one metal and one nonmetal available.

[b] The prototypes that has either only metals or only nonmetals were grouped as a mixed 'prototype'.

number of mathematical operators in the features). However, we note that many prototypes have very few data points and therefore the classification model risks to be overfit.

MT-SISSO mediates between the two extrema of the global, inaccurate map and the one-per-prototype map, that is probably overfit for prototypes for which few data points are available. Interpreting the map for one prototype as one task, MT-SISSO can be set up to look for a set of maps, all defined by the same descriptor, but with differently located convex regions for the classification. We ran MT-SISSO for classification with the same parameter settings as for the global descriptor, except that the prototype ReO$_3$ is excluded (this prototype is represented by only 1 metal and 1 nonmetal in our reference dataset) and the crystal features x_A, x_B, $N_A^{\rm N}$, and $N_B^{\rm N}$ are removed because they are constant within a given prototype. Figure 10 shows the MT-SISSO maps. Overall and individually, they achieve perfect classification. The common 2D descriptor is:

$$\begin{aligned} d_1 &= \frac{V_{\rm cell}}{\sum V_{\rm atom}}\frac{\chi_B\exp(r_{{\rm cov},A})}{\chi_A r_{{\rm cov},A}} \\ d_2 &= \frac{V_{\rm cell}}{\sum V_{\rm atom}}\mathrm{IE}_A\mathrm{IE}_B r_{{\rm cov},A}\sqrt{\chi_A/\exp(\chi_A)}. \end{aligned} \tag{7}$$

We note that this descriptor has similar 'ingredients' (primary features) as the global ST-SISSO descriptor presented in [25], in particular the descriptor depends linearly on the inverse of the packing fraction $\sum V_{\rm atom}/V_{\rm cell}$, which is the only selected collective feature, i.e. related to the actual atomic structure of the material.

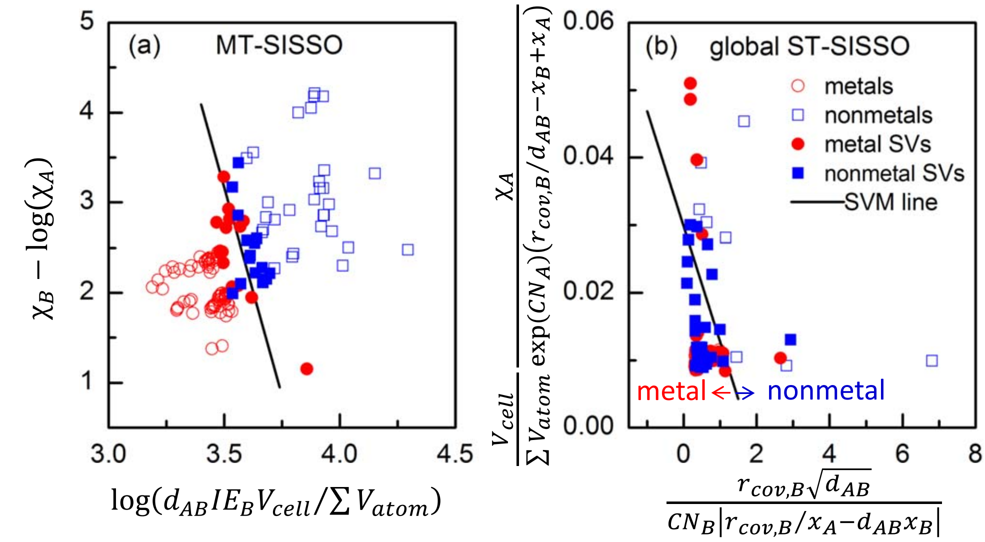

Figure 11. 'Leave-rock-salt-prototype-out' generalization test for MT-SISSO (left panel) and global ST-SISSO (right panel). Both MT-SISSO and ST-SISSO descriptors are trained on all prototypes except rock-salt and the support-vector-machine (SVM) line is trained on the rock-salt data, at fixed descriptor. The filled symbols are the support vectors (SV) of the SVM model.

On the same machine as for the octet-binary application, the learning fro producing figure 10 required about 1 h runtime.

To demonstrate the generalizability of MT-SISSO descriptors on unseen prototype materials, we performed a 'leave-one-prototype-out' validation. In practice, we focused on the RS prototype (that includes about 40% of the training dataset) and we trained the metal/nonmetal classification with MT-SISSO and with global ST-SISSO. The latter is ST-SISSO by using all training data to train a single metal/nonmetal map. This is the same approach as in [25], where however fewer prototypes were considered. For ST-SISSO, the features coordination number N^{N} and atomic fraction x are included as primary features in $\mathbf{\Phi}_0$. Subsequently the RS data are projected into the 2D descriptor determined by the training on the other prototypes and a SVM model is trained at fixed descriptor. We name these two approaches MT-SISSO+SVM and ST-SISSO+SVM. In this test, we have omitted the ST-SISSO learning on one prototype because all the data points of the left-out prototype are left out of training at the SISSO stage. The results are shown figure 11. The descriptor identified by global ST-SISSO scatters metals and nonmetals NaCl binaries all around the map, making a classification impossible. In contrast, the MT-SISSO descriptor yields a map that separates fairly metals versus nonmetals, without having access to any direct information on RS materials in the training. Quantitatively, the number of misclassified NaCl materials by MT-SISSO+SVM is 6 out of 132 and one can appreciate by naked eye in figure 11(a) that the misclassification is not 'severe', i.e. the misclassified materials are close to the SVM line. For ST-SISSO+SVM the number of misclassified materials is 36 out of 132 and visual inspection (figure 11(b)) reveals that, without the labels 'metal' ('nonmetal') in the half planes, it would be even difficult to decide which side of the line is predicted to contain metals (nonmetals).

We repeated the test for other prototypes, but, mainly due to the fact that they individually contain far less data than RS, the comparison between MT- and ST-SISSO is less insightful. We nonetheless report the result in the supplementary material, which is available online at stacks.iop.org/JPMATER/2/024002/mmedia.

4. Conclusions

In conclusion, we have introduced a nontrivial extension of the SISSO algorithm. Such an extension is called MT-SISSO, it belongs to the wider class of learning algorithms known as MT Learning, and is specifically designed for learning from databases with randomly or selectively distributed missing information. MT-SISSO finds a common descriptor, in terms of analytical functions of simple input physical quantities called *primary features*, when learning different properties (tasks) simultaneously. This joint learning yields robust models also with large amount of missing data, as demonstrated with two showcase materials-science examples: the prediction of the ground-state crystal structure for octet binaries compounds (out of 5 candidate structures) and the prediction of metal versus nonmetal classification of binary materials distributed over 17 crystal structure prototypes. Since materials databases typically contain data from different sources and therefore unsystematic (different properties are collected for different materials), MT-SISSO is a method that can be suitably applied to these databases to yield predictive models for properties of interest.

The ST- and MT-SISSO package, as used for obtaining the results presented in this paper, is maintained by R Ouyang and available open access at github.com/rouyang2017/SISSO.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program (#676580: The NOMAD Laboratory—an European Center of Excellence and #740233: TEC1p), the Berlin Big-Data Center (BBDC, #01IS14013E), and BiGmax, the Max Planck Society's Research Network on Big-Data-Driven Materials-Science.

ORCID iDs

Luca M Ghiringhelli https://orcid.org/0000-0001-5099-3029

References

[1] Office of Science and Technology Policy, White House 2011 Materials Genome Initiative for Global Competitiveness https://obamawhitehouse.archives.gov/mgi
[2] Curtarolo S, Hart G L W, Setyawan W, Mehl M J, Jahnátek M, Chepulskii R V, Levy O and Morgan D 2010 AFLOW: software for high-throughput calculation of material properties http://materials.duke.edu/aflow.html
[3] Jain A, Hautier G, Moore C J, Ong S P, Fischer C C, Mueller T, Persson K A and Ceder G 2011 *Comput. Mater. Sci.* **50** 2295
[4] Saal J E, Kirklin S, Aykol M, Meredig B and Wolverton C 2013 *JOM* **65** 1501
[5] Landis D D, Hummelshøj J, Nestorov S, Greeley J, Dułak M, Bligaard T, Nørskov J K and Jacobsen K W 2012 *Comput. Sci. Eng.* **14** 51
[6] Ghiringhelli L M, Carbogno C, Levchenko S, Mohamed F, Huhs G, Lüders M, Oliveira M and Scheffler M 2017 *NPJ Comput. Mater.* **3** 46
[7] Draxl C and Scheffler M 2018 *MRS Bull.* **43** 676
[8] Draxl C and Scheffler M 2018 Big-data-driven materials science and its fair data infrastructure *Handbook of Materials Modeling* ed S Yip and W Andreoni (Berlin: Springer)
[9] Hey T, Tansley S and Tolle K 2009 *The Fourth Paradigm: Data-Intensive Scientific Discovery* (Redmond, WA: Microsoft Research)
[10] Bartók A, Albert P, Payne M C, Kondor R and Csányi G 2010 *Phys. Rev. Lett.* **104** 136403
[11] Carrete J, Mingo N, Wang S and Curtarolo S 2014 *Adv. Funct. Mater.* **24** 7427
[12] Rajan K 2015 *Annu. Rev. Mater. Res.* **45** 153
[13] Mueller T, Kusne A G and Ramprasad R 2016 Machine learning in materials science *Reviews in Computational Chemistry* (New York: Wiley) pp 186–273
[14] Kim C, Pilania G and Ramprasad R 2016 *Chem. Mater.* **28** 1304
[15] Faber F A, Lindmaa A, von Lilienfeld O A and Armiento R 2016 *Phys. Rev. Lett.* **117** 135502
[16] Takahashi K and Tanaka Y 2016 *Dalton Trans.* **45** 10497
[17] Bartók A, De S, Poelking C, Bernstein N, Kermode J, Csányi G and Ceriotti M 2017 *Sci. Adv.* **3** 1701816
[18] Goldsmith B R, Boley M, Vreeken J, Scheffler M and Ghiringhelli L M 2017 *New J. Phys.* **19** 013031
[19] Pham T L, Nguyen N D, Nguyen V D, Kino H, Miyake T and Dam H C 2018 *J. Chem. Phys.* **148** 204106
[20] Ghiringhelli L M, Vybiral J, Levchenko S V, Draxl C and Scheffler M 2015 *Phys. Rev. Lett.* **114** 105503
[21] Candès E J, Romberg J and Tao T 2006 *IEEE Trans. Inf. Theory* **52** 489
[22] Donoho D L 2006 *IEEE Trans. Inf. Theory* **52** 1289
[23] Nelson L J, Hart G L, Zhou F and Ozoliņš V 2013 *Phys. Rev.* B **87** 035125
[24] Ghiringhelli L M, Vybiral J, Ahmetcik E, Ouyang R, Levchenko S V, Draxl C and Scheffler M 2017 *New J. Phys.* **19** 023017
[25] Ouyang R, Curtarolo S, Ahmetcik E, Scheffler M and Ghiringhelli L M 2018 *Phys. Rev. Mater.* **2** 083802
[26] Fan J and Lv J 2008 *J. R. Stat. Soc.* B **70** 849
[27] Tibshirani R 1996 *J. R. Stat. Soc.* B **58** 267
[28] Tropp J A and Gilbert A C 2007 *IEEE Trans. Inf. Theory* **53** 4655
[29] Pati Y C, Rezaiifar R and Krishnaprasad P S 1993 *The 27th Asilomar Conf.: Signals, Systems and Computers* vol 1 (Pacific Grove, CA: IEEE) pp 40–4
[30] Bartel C J, Sutton C, Goldsmith B R, Ouyang R, Musgrave C B, Ghiringhelli L M and Scheffler M 2019 *Sci. Adv.* **5** eaav0693
[31] Bartel C J, Millican S L, Deml A M, Rumptz J R, Tumas W, Weimer A W, Lany S, Stevanović V, Musgrave C B and Holder A M 2018 *Nat. Commun.* **9** 4168
[32] Caruana R 1997 *Mach. Learn.* **28** 41
[33] Obozinski G, Taskar B and Jordan M 2006 Multi-task feature selection *Tech. Rep.* Department of Statistics, University of California, Berkeley
[34] Argyriou A, Evgeniou T and Pontil M 2008 *Mach. Learn.* **73** 243
[35] Yin X and Liu X 2018 *IEEE Trans. Image Process.* **27** 964
[36] Gong P, Ye J and Zhang C 2013 *J Mach. Learn. Res.* **14** 2979
[37] Huang B, Ke D, Zheng H, Xu B, Xu Y and Su K 2015 *Proc. INTERSPEECH (Dresden, Germany)* pp 2464–8
[38] Thung K-H and Wee C-Y 2018 *Multimed. Tools Appl.* **77** 29705
[39] Zhang Y and Yang Q 2018 *Natl. Sci. Rev.* **5** 30
[40] Pilania G, Gubernatis J E and Lookman T 2017 *Comput. Mater. Sci.* **129** 156
[41] Ahmetcik E and Ziletti A 2017 https://analytics-toolkit.nomad-coe.eu/hands-on-cs
[42] Mohamed F, Kariryaa A, Ziletti A, Ahmetcik E, Ghiringhelli L M and Scheffler M 2017 https://analytics-toolkit.nomad-coe.eu

[43] Regler B 2017 https://analytics-toolkit.nomad-coe.eu/tutorial-periodic-table

[44] Blum V, Gehrke R, Hanke F, Havu P, Havu V, Ren X, Reuter K and Scheffler M 2009 *Comput. Phys. Commun.* **180** 2175

[45] Ahmetcik E 2016 Machine learning of the stability of octet binaries *Master's Thesis* Fritz-Haber-Institut der Max-Planck-Gesellschaft, Berlin (https://th.fhi-berlin.mpg.de/site/uploads/Publications/Masterthesis_AhmetcikEmre.pdf)

[46] SpringerMaterials database https://materials.springer.com/

[47] WebElements https://webelements.com