Science Bulletin 68 (2023) 1259–1270

Contents lists available at ScienceDirect

Science Bulletin

journal homepage: www.elsevier.com/locate/scib

Article

Auto-MatRegressor: liberating machine learning alchemists

Yue Liu [a,b,e], Shuangyan Wang [a], Zhengwei Yang [a], Maxim Avdeev [f,g], Siqi Shi [c,d,e,*]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[b] Shanghai Engineering Research Center of Intelligent Computing System, Shanghai 200444, China
[c] State Key Laboratory of Advanced Special Steel, School of Materials Science and Engineering, Shanghai University, Shanghai 200444, China
[d] Materials Genome Institute, Shanghai University, Shanghai 200444, China
[e] Zhejiang Laboratory, Hangzhou 311100, China
[f] Australian Nuclear Science and Technology Organisation, Sydney 2232, Australia
[g] School of Chemistry, The University of Sydney, Sydney 2006, Australia

ARTICLE INFO

Article history:
Received 30 November 2022
Received in revised form 25 February 2023
Accepted 8 May 2023
Available online 22 May 2023

Keywords:
Materials property prediction
Machine learning
Automatic modeling
Meta-learning

ABSTRACT

Machine learning (ML) is widely used to uncover structure–property relationships of materials due to its ability to quickly find potential data patterns and make accurate predictions. However, like alchemists, materials scientists are plagued by time-consuming and labor-intensive experiments to build high-accuracy ML models. Here, we propose an automatic modeling method based on meta-learning for materials property prediction named Auto-MatRegressor, which automates algorithm selection and hyperparameter optimization by learning from previous modeling experience, i.e., meta-data on historical datasets. The meta-data used in this work consists of 27 meta-features that characterize the datasets and the prediction performances of 18 algorithms commonly used in materials science. To recommend optimal algorithms, a collaborative meta-learning method embedded with domain knowledge quantified by a materials categories tree is designed. Experiments on 60 datasets show that compared with the traditional modeling method from scratch, Auto-MatRegressor automatically selects appropriate algorithms at lower computational cost, which accelerates constructing ML models with good prediction accuracy. Auto-MatRegressor supports dynamic expansion of meta-data with the increase of the number of materials datasets and other required algorithms and can be applied to any ML materials discovery and design task.

1. Introduction

Materials scientists are constantly striving to boost design and discovery of novel materials with superior properties. Recently, data-driven machine learning (ML) has been receiving increasing attention due to its ability to quickly and accurately find relationships between materials properties and complex factors [1–3]. A large amount of materials raw data has been accumulated in some databases (e.g., Materials Project (MP) [4], AFLOWlib [5], and Open Quantum Materials Database (OQMD) [6]), thereby providing a good data foundation for ML research, such as Matbench [7]. The usage of ML covers various prediction tasks of continuous properties, such as band gap, formation energy, thermodynamic stability, ionic conductivity, and mechanical properties.

The researchers in materials science using ML aim to make predictions of materials properties by constructing structured data based on raw data to reveal the "Composition-Structure-Process-Property" relationships. However, construction of high-accuracy ML models is challenging. Herein, we summarize the ML models employed in 78 selected publications on materials research, shown in Fig. 1a. According to the "no free lunch" theorem [8,9], there is no single best ML algorithm for all materials problems. Therefore, to tackle a new materials problem, researchers need to find the best model from many available models. Generally, it involves testing performance of multiple ML algorithms to select the most suitable one [10–13].

In addition, the performance, training speed, and complexity of the ML model are sensitive to its hyperparameters [14,15]. Like alchemists, materials researchers often determine the optimal hyperparameters through labor-intensive tuning or drawing on historical experience in materials science using ML [16,17]. As shown in Fig. 1b, the hyperparameters vary for different ML algorithms, which exacerbates the problem since they also need to be tuned to construct the optimal ML model for a specific materials problem. To simplify this process, some swarm intelligence

* Corresponding author.
E-mail address: sqshi@shu.edu.cn (S. Shi).

https://doi.org/10.1016/j.scib.2023.05.017
2095-9273/

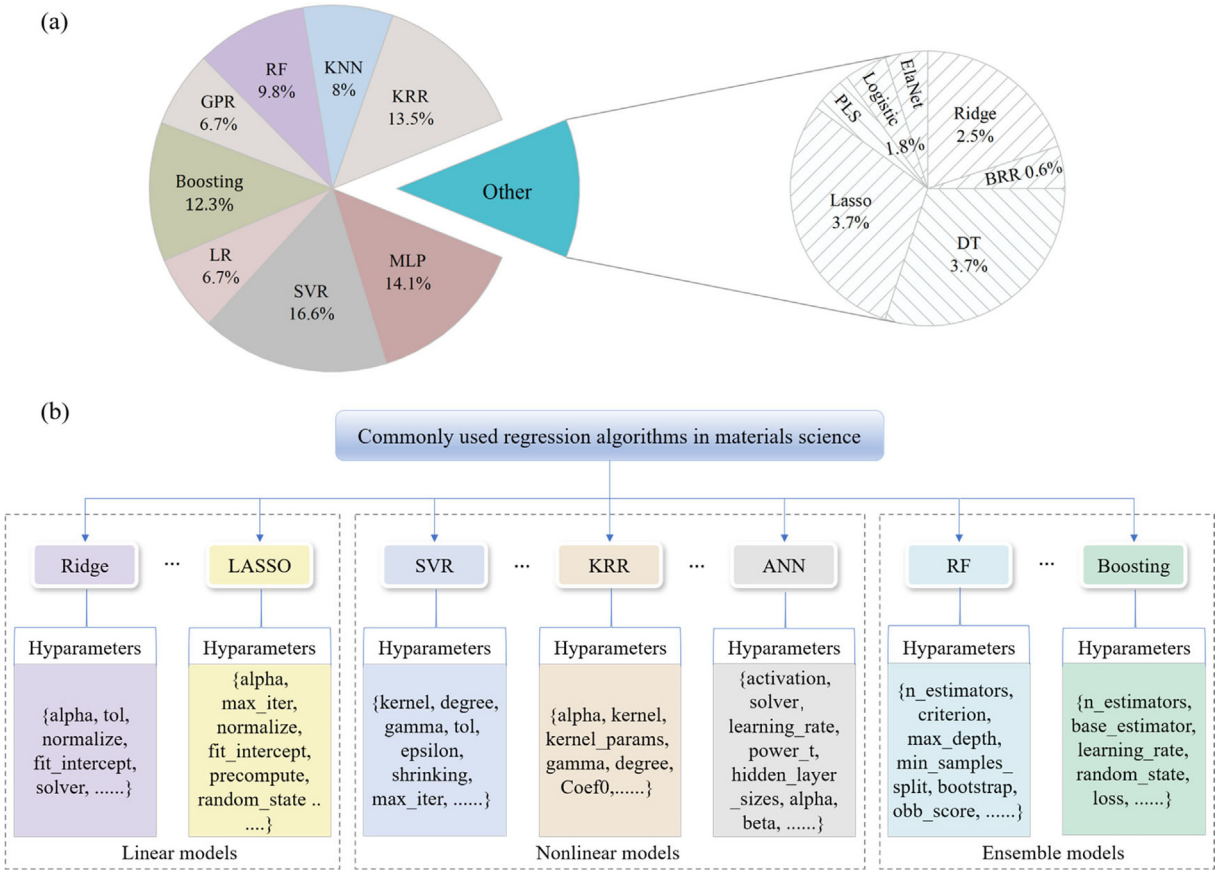

Fig. 1. (Color online) The commonly used regression algorithms in materials science. (a) The usage frequency of different algorithms in 78 publications. The ensemble models, including Random Forest (RF) and Boosting models, and the nonlinear models, including Multi-Layer Perception (MLP), K-Nearest Neighbor (KNN), Support Vector Regression (SVR), Gaussian Process Regression (GPR), Decision Tree (DT), and Kernel Ridge Regression (KRR), account for the top few, while the proportions of other linear models, including Ridge Regression (Ridge), Logistic Regression (Logistic), Linear Regression (LR), Bayesian Ridge Regression (BRR), Lasso Regression (Lasso), Elastic Net Regression (ElaNet), and Partial Least Squares (PLS), are not high. (b) The complex hyperparameters of different algorithms.

algorithms, such as ant colony optimization [18], particle swarm optimization [19], and genetic algorithm [20], have been adopted to explore hyperparameter space. All that increases the computational cost of constructing a high-quality predictive model [21–23]. Therefore, it is a burning issue to develop a novel automatic method for selecting regression models and optimizing their hyperparameters more quickly and efficiently, to improve the usability and reliability of ML in materials science.

Automated machine learning (AutoML) offers a path to solve this issue, which supports building ML models automatically without extensive ML knowledge and massive human intervention. Several efforts, such as Auto-Weka [24], the tree-based pipeline optimization technique (TPOT) [25], and Hyperopt-Sklearn [26], have achieved remarkable results in effectively improving the usability and accuracy of ML models. However, these end-to-end modeling processes to build high-quality ML models have high computational costs and are often overwhelmingly technically involved for practitioners. As a solution, Dunn et al. [7] proposed Automatminer, which applies the TPOT technique to reduce the intervention of materials experts in the modeling process and then predict the physicochemical properties of inorganic bulk materials. To further develop AutoML, the relationship between materials datasets and ML algorithms is being explored by employing the idea of meta-learning. The goal of introducing meta-learning into AutoML is to correlate the performance of ML algorithms with data characteristics of previous tasks [27–30], i.e., meta-features, which are collectively referred to as meta-data. This enables to warm-start model construction of any new task by automatically recommending the promising algorithms and their corresponding hyperparameters based on modeling experience on historical tasks.

Here, we propose an automatic modeling method based on meta-learning for materials property prediction, named Auto-MatRegressor. For the approach to work, there are two main challenges: (1) what meta-data should be constructed; (2) how to resolve the overfitting for limited materials datasets.

Towards the first challenge, implementing suitable meta-features is a fundamental issue for determining which modeling experience can be learned from, and there have been lots of excellent efforts to address it with many meta-features in the clustering or classification studies [31,32]. To comprehensively characterize the materials datasets used for regression tasks, we implement a total of 27 meta-features including 24 traditional meta-features and 3 enhanced meta-features. In addition, we employ 18 regression algorithms commonly used in materials science as candidate algorithms to collect their prediction performance on all previous datasets.

Towards the second challenge, sufficiently large datasets are required for the meta-learning-based AutoML model to avoid overfitting and improve the accuracy of algorithm recommendations [33]. However, in materials science, only limited datasets are often available. One of our solutions is to introduce regression datasets from other research fields (called "general datasets") as part of experimental data. Furthermore, applying data-driven modeling without the guidance of materials experts often leads to the results of meta-learning being inconsistent or even contradictory to materials domain knowledge. As a solution, we design a collaborative

meta-learning model embedded with domain knowledge to efficiently combine the meta-learning results from the materials datasets with those from the general datasets. Without requiring materials experts to have strong ML knowledge and conduct labor-intensive experiments, our method can automatically and quickly build an ML model with high prediction performance for a new materials task. The efficacy and robustness of Auto-MatRegressor are demonstrated on a variety of experimental datasets from different materials categories.

2. Methods

2.1. The workflow of Auto-MatRegressor

As shown in Fig. 2, the general workflow of applying ML to the field of materials science involves five major stages, starting from target definition to data processing, feature engineering, model construction, and finally application. During the model construction, materials experts often rely on trial-and-error testing to find the ML model with good prediction performance. It is recognized that this stage requires materials experts' intervention and is relatively expensive in terms of human (researcher) time and expertise [34,35]. Herein, an automatic regression method for the materials property prediction named Auto-MatRegressor is proposed to automate the model construction process.

Auto-MatRegressor is composed of three procedures. (1) Constructing meta-data for supporting meta-learning. To avoid overfitting, 54 materials datasets from the materials ML publications and 60 general datasets screened from ML online data repositories are collected manually as the overall experimental data. Further details of the datasets can be found in Tables S1, S2 and the Supplementary materials Note 1 (online). Then, based on these two types of datasets, two different sets of regression-oriented meta-data are constructed, respectively. (2) Building a collaborative meta-learning model for promising algorithms. A multinode tree is designed and constructed to visualize domain knowledge involving different materials categories, named "Materials Categories Tree" (MCT-DK). Then, based on the various meta-data, meta-learning is guided by MCT-DK to collaboratively recommend two distinct rankings of regression algorithms. (3) Optimizing hyperparameters for the recommended regressors. The Bayesian optimization (BO) algorithm is employed to search the global optimal hyperparameters (the Supplementary materials Note 3).

As a result, given a new dataset, Auto-MatRegressor can automatically recommend well-promising ML algorithms by learning the experience of building models on similar datasets, which are then applied to predictions of unknown target properties.

2.2. Meta-data construction

Meta-data is composed of the algorithm performance and meta-features that characterize prior datasets. Meta-features determine the similarity measure between datasets, which in turn affects the recommendation of ML algorithms for new tasks. Herein, the prior datasets including 54 materials datasets and 60 general datasets are used to support the construction of meta-data. The materials datasets are sourced from various subdisciplines of materials sciences, such as lithium superionic conductor properties, organic compound properties, and biofuel compound properties. The number of samples ranges from 9 to 7230, representing both relatively scarce and relatively abundant properties.

2.2.1. Meta-features

Meta-features aim to characterize datasets to identify similarities and differences, and then pick up similar datasets from existing ones for a new task. All meta-features adopted in this paper are shown in Table 1. The traditional meta-features [30,37] include simple meta-features, statistical meta-features, principal component analysis-based (PCA-based) meta-features, and landmarking meta-features. The PCA-based meta-features are extracted from the condition attributes of the datasets; the landmarking meta-features that depend on ML algorithms including Lasso, KNN, SVR, MLP, and DT, are extracted from model performance measures. Additionally, the construction of a purely data-driven ML model is usually done under the assumption that learning samples conform to a certain data distribution [14,38]. Also, the target

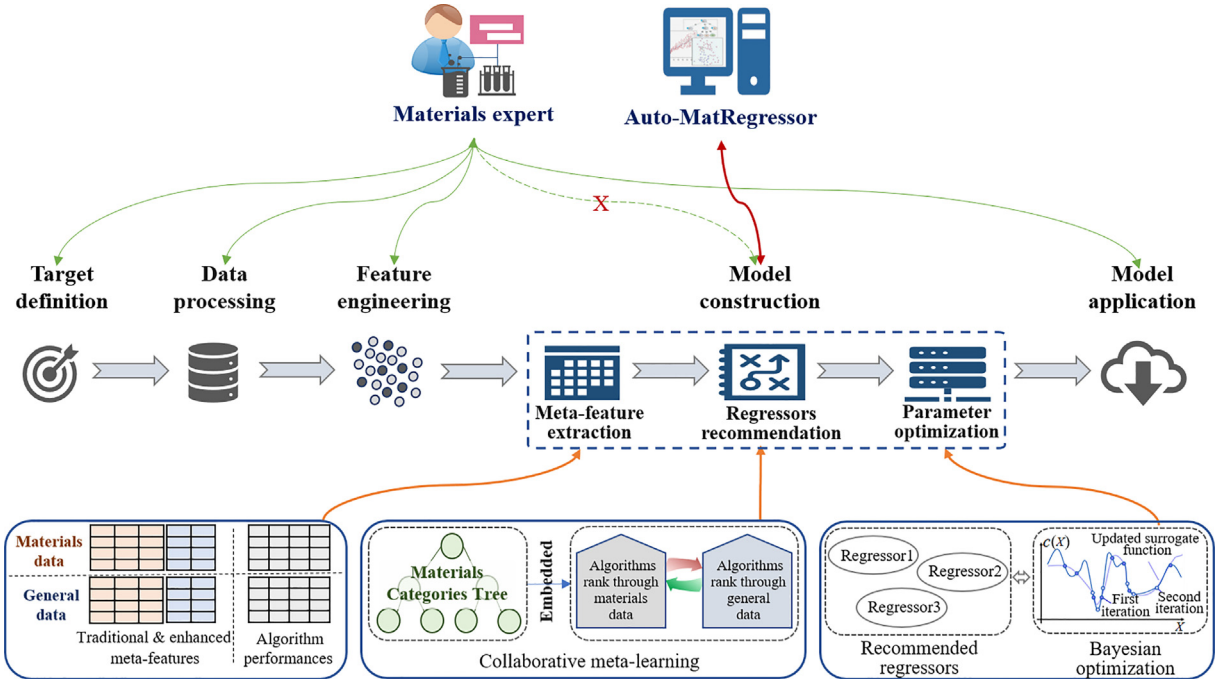

Fig. 2. (Color online) Automatic model construction for the materials property prediction (Auto-MatRegressor).

Table 1
List of meta-features in Auto-MatRegressor.

	Category	Meta-features	Description
Traditional meta-features	Simple[a]	Number of instances, Log number of instances, Number of features, Log number of features, Dataset ratio, Log dataset ratio, Inverse dataset ratio, Inverse log dataset ratio	The simple meta-features are directly extracted from the data and represent basic information about the dataset. It is generally advisable to transform original attributes by taking logarithms, ratios, or inverse to make the marginal distribution as near normal as possible [36]. They are the simplest set of measures in terms of definition and computational cost.
	Statistical[a]	Kurtosis max, Kurtosis min, Kurtosis mean, Kurtosis std, Skewness min, Skewness max, Skewness mean, Skewness std	The statistical meta-features capture data distribution. Skewness is a measure of symmetry in distribution to describe how much statistical data distribution is asymmetrical from the normal distribution. Kurtosis is the measure of heaviness or the density of distribution tails to determine whether a distribution contains extreme values.
	PCA-based[a]	PCA 95%, PCA kurtosis first pc, PCA skewness first pc	The PCA-based meta-features perform principal component analysis and compute various statistics of the principal components based on intrinsic dimensionality.
	Landmarking[b]	Lasso-based landmarking, KNN-based landmarking, SVR-based landmarking, MLP-based landmarking, DT-based landmarking	The landmarking meta-features characterize datasets by using the performance of a set of simple and fast learning algorithms (together with preset hyperparameter values) with different inductive biases to capture relevant information with a low computational cost. The metric root mean square error (RMSE) is used for evaluating the predictive performance.
Enhanced meta-features	Target property-based statistical[c]	Kurtosis-*Y*, Skewness-*Y*	The target property-based statistical meta-features are extracted to characterize the output-data distribution via descriptive statistics.
	Uncertainty-based[c]	(*Ey*, *En*, *He*)	The uncertainty-based meta-feature includes three numerical characteristics: (a) expectation *Ey*; (b) entropy *En*; (c) and hyper entropy *He*.

[a] The simple meta-features, statistical meta-features, and PCA-based meta-features are extracted from the condition attributes of the datasets.
[b] The landmarking meta-features depend on ML algorithms to extract model performance measures.
[c] The target property-based statistical meta-features and uncertainty-based meta-feature are extracted from the target properties of the datasets. Data distribution and uncertainty associated with target properties are crucial to characterize or distinguish different materials datasets in meta-learning for regression tasks.

properties are often affected by various physicochemical factors due to complex driving mechanisms in materials research. Experimental errors from measurements or computational errors from incorrect approximations can lead to uncertainty in the target properties of the collected materials data [9,39–41]. Therefore, data distribution and uncertainty associated with target properties are crucial for characterizing or distinguishing different materials datasets in meta-learning for regression tasks. To address the issue, in this work, we introduce two types of enhanced meta-features associated with target properties. The variables of the meta-features implemented in Auto-MatRegressor are shown in the Supplementary materials Note 3 and Table S5 (online).

2.2.2. Model performance

Building on meta-features, promising ML algorithms stored in the previously collected datasets can be recommended for a new given dataset. It is crucial for meta-learning to build high-performance predictive models based on prior datasets. In terms of prediction accuracy, classical and statistical ML approaches (e.g., LR, SVR, KNN, and DT) are more suitable for smaller datasets; neural networks require larger amounts of data and only become feasible for datasets with the number of points on the order of thousands or more. From the perspective of model interpretability, the linear models (e.g., LR, PLS, and Ridge) are simple to implement, and the learning results tend to be comprehensible. However, the relationships between the conditional factors and the target attribute in materials science are often complex, which leads to the fact that although the learning results of the nonlinear models (e.g., MLP, GPR, and SVR) are "black boxes", they are widely used by materials experts. Combined with the analysis of usage frequency of regression algorithms (Fig. 1), 18 regression algorithms are considered in Auto-MatRegressor: LR, Lasso, Ridge, LassoLars (LS), BRR, Random Sample Consensus (RANSAC), KNN, GPR, DT, Adaptive Boosting (AdaBoost), RF, MLP, Stochastic Gradient Descent (SGD), SVR, Linear SVR, KRR, ElaNet, and Passive Aggressive Regression (PAR). Each experimental dataset is divided into 80% training samples for prediction model development and 20% test samples for model performance evaluation. Herein, we tune the hyperparameters via the 5-fold cross-validation (CV) and BO [42] method. As shown in Eq. (1), RMSE is used for evaluating the predictive performance, which is efficiently implemented in a Python programming language [24]. Finally, we successfully design and store high-performing regressors for each dataset by Auto-MatRegressor. The details of hyperparameters are presented in the Supplementary materials.

$$\mathrm{RMSE}\left(\boldsymbol{Y},\widehat{\boldsymbol{Y}}\right)=\sqrt{\frac{\sum_{i=1}^{n}\left(y_i-y_i'\right)^2}{n}}, \tag{1}$$

where n is the total number of samples used for training ML models; y_i and y_i' represent the real value and predicted value, respectively.

2.3. Collaborative meta-learning embedded with materials domain knowledge

2.3.1. Domain knowledge acquisition and quantification

Exploratory works in our research group have demonstrated that embedding domain knowledge into the modeling process is

an effective strategy to enhance the reliability and interpretability of ML results [1,43,44]. Here, MCT-DK is designed to visualize materials domain knowledge. The target properties of materials datasets are first considered as the leaf nodes. Then, materials experts design the penultimate layer by analyzing the materials categories to which the predicted attributes belong. Likewise, the other layers are designed through this same strategy until reaching the root of the tree. Although some branches (materials categories) lack the corresponding leaf nodes (datasets), materials experts keep these branches to ensure the integrity of the materials categories system. The design of MCT-DK is illustrated in Fig. 3a, where the detailed representation of leaf nodes is presented in the Supplementary materials Note 2 and Table S4 (online).

From the perspective of physical and chemical properties (as shown in Fig. 3b), materials are classified under four categories of (1) metal; (2) inorganic non-metallic; (3) polymer, and (4) composite. Metal materials contain ten materials datasets concerning multiple subdivided materials such as Fe-based metallic glasses, Ni-based single crystal superalloys, and high entropy alloys. Inorganic non-metallic materials consist of five basic groups: glass, ceramic materials, cement, refractory materials, and others that are difficult to categorize explicitly into any of the above subclasses. Notably, there are thirty-two datasets across the targeted properties of Ge_xSe_{1-x} glass, lithium superionic conductors, oxide ionic conductors, etc., which can be attributed to more research focusing on inorganic non-metallic materials. Polymer materials are divided into seven categories such as plastic, rubber, and fiber. Unlike the above three kinds of materials, to the best of our knowledge, there is less publicly available data on composite materials. The MCT-DK is encoded and quantified by the weight score s_i^{mat} to measure the similarity between the materials datasets at the domain knowledge level, as shown in Fig. 3c. The implementation of s_i^{mat} is detailed in the Supplementary materials Note 2 (online).

2.3.2. Collaborative recommendation embedded with MCT-DK

Firstly, two different types of meta-data are constructed, corresponding to materials datasets and general datasets, respectively. Given a new dataset, 27 regression-oriented meta-features are first extracted by Auto-MatRegressor. Then, the Euclidean distance between the new dataset and each prior dataset is calculated in

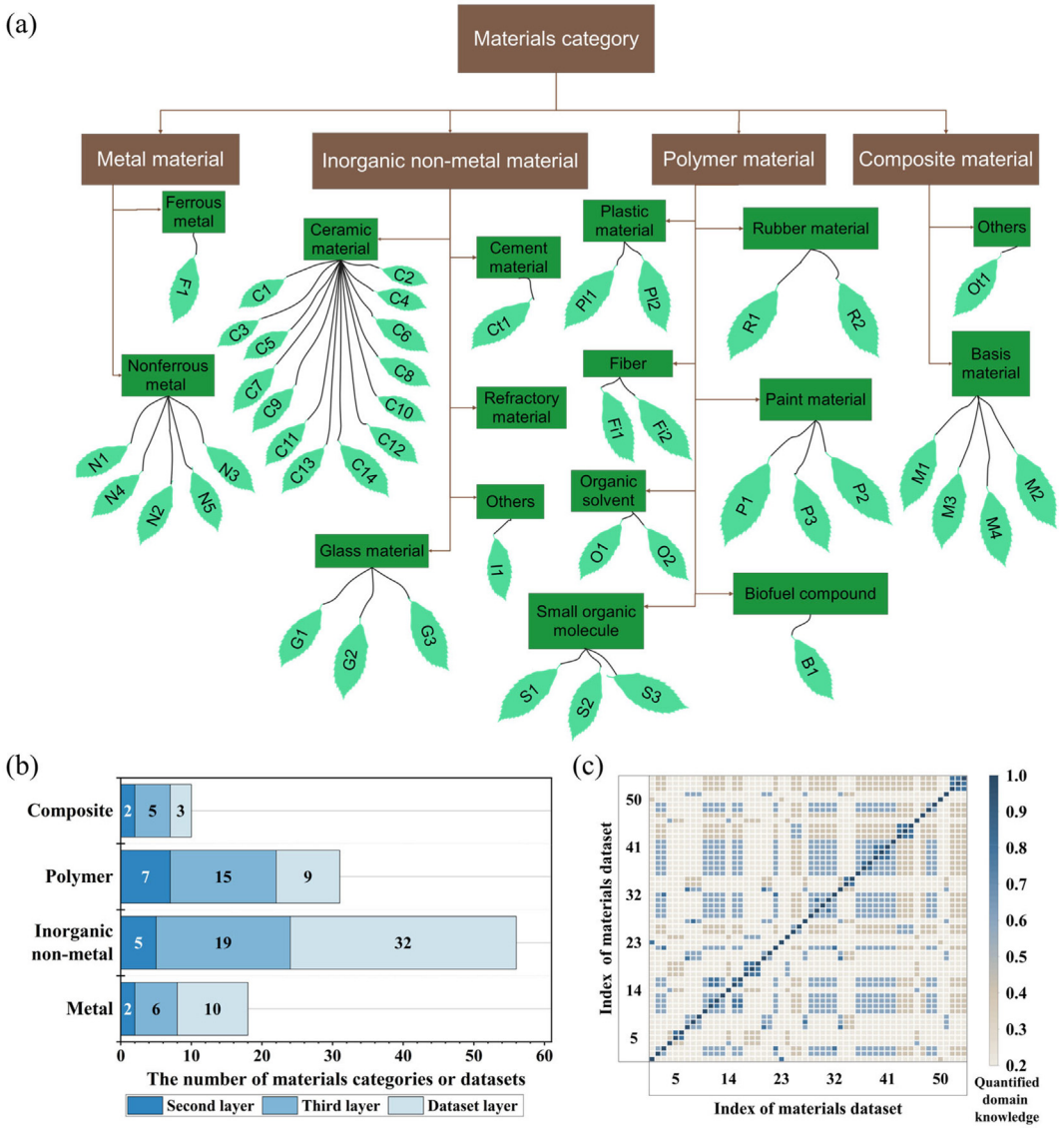

Fig. 3. (Color online) Quantified domain knowledge based on MCT-DK. (a) MCT-DK to visualize domain knowledge. (b) The data distribution corresponding to every layer of MCT-DK. Numbers in the bars represent the number of materials categories or datasets. (c) The similarity heatmap at domain expert knowledge level. The deeper the color of the cell in the heatmap, the more similar between every two datasets.

meta-features space to identify similar datasets and the corresponding algorithm performance from meta-data. To formally define the problem, we denote $F_i = \{f_{i,1}, \cdots, f_{i,k}, \cdots, f_{i,n}\}$ to be a set of meta-features indexed by k, where n denotes the number of meta-features. The distance measured between F_i and F_j is given by Eq. (2):

$$d(F_i, F_j) = d_{i,j} = \sqrt{\sum_{k}^{n} \left(f_{i,k} - f_{j,k}\right)^2}. \tag{2}$$

Subsequently, all prior datasets are ranked according to the Euclidean distance. In terms of algorithm performance from similar general datasets, the Average Ranking (AR) method is used to return the final rankings of regression algorithms. It works as follows.

Let $r_j = (\overline{p_{1,j}}, \cdots, \overline{p_{i,j}}, \cdots, \overline{p_{g,j}})$ be the rank position of algorithm j ($j = 1, \cdots, a$) where a is the number of regression algorithms indexed by j, and g is the number of similar general datasets indexed by i. The average rank position $\bar{r}_j$ for any regression algorithm is given by:

$$\overline{r_j} = \frac{\sum_{i=1}^{g} \overline{p_{i,j}}}{g}, \tag{3}$$

where $\overline{p_{i,j}}$ represents the ranking of the j-th regression algorithm for the i-th similar dataset in the case of the evaluation index RMSE.

After calculating the rankings $\overline{p_{i,j}}$, the AR method embedded with materials domain knowledge is improved as follows $\overline{r_j^{\mathrm{DK}}}$:

$$\overline{r_j^{\mathrm{DK}}} = \frac{s_1^{\mathrm{mat}}\overline{p_{1,j}} + \cdots + s_i^{\mathrm{mat}}\overline{p_{i,j}} + \cdots + s_m^{\mathrm{mat}}\overline{p_{m,j}}}{m} = \frac{\sum_{i=1}^{m} s_i^{\mathrm{mat}}\overline{p_{i,j}}}{m}, \tag{4}$$

where m is the number of similar materials datasets indexed by i. s_i^{mat} represents the similarity score between every two materials datasets according to MCT-DK.

According to Eqs. (3) and (4), recommendation rankings $R^{\mathrm{gd}} = \{\overline{r_1}, \cdots, \overline{r_j}, \cdots, \overline{r_a}\}$ and $R^{\mathrm{md}} = \left\{\overline{r_1^{\mathrm{Dk}}}, \cdots, \overline{r_j^{\mathrm{Dk}}}, \cdots, \overline{r_a^{\mathrm{Dk}}},\right\}$ are calculated. Finally, we propose a novel strategy to effectively collaborate the above two results, as defined in Eq. (5):

$$col_{R\left(R^{\mathrm{md}}, R^{\mathrm{gd}}, C^*\right)} = C^* R^{\mathrm{md}} + (1 - C^*) R^{\mathrm{gd}}, \tag{5}$$

where C^* represents the weight for the rank position of each regression algorithm indexed by j. Additionally, the specific values of C^* varies with the different new datasets for the optimal value in the interval $[0.1, 1]$. After the optimal collaborative ranking of each regression algorithm is found, the ultimate rankings are recommended by rearranging the collaborative rankings.

2.4. Evaluation indices for ranking accuracy

To assess the quality of the recommended rankings, Spearman's rank correlation (SRC) [45] is used to measure the similarity between recommended ranking and true ranking, as defined in Eq. (6). The true ranking is implemented by the Combined Algorithm Selection and Hyperparameter (CASH) [24], in which the regression algorithm selection is viewed as a super-hyperparameter and executed with hyperparameter optimization simultaneously.

$$SRC(rr, rt) = 1 - \frac{6\sum_{j=1}^{a}\left(rr_j - rt_j\right)^2}{a^3 - a}, \tag{6}$$

where rr and rt are the recommended ranking and the true ranking, respectively; a is the number of all the algorithms with rank positions indexed by j. The coefficient ranges from $[-1, +1]$, whereby the larger the value of SRC, the higher the similarity between the true and the recommended rankings. The SRC has been used for ranking comparison in several meta-learning works [27,46,47].

Additionally, given a recommended ranking of all the candidate algorithms, it is reasonable to expect that the algorithm ranked at the top is considered first, followed by the one ranked second, and so on. The better the quality of the recommendation found by the algorithm, the higher its ranking position as well as the possibly better its fitting. Therefore, to better illustrate the reliability of the recommended result, the recall of the top-N algorithms is proposed in this paper as another evaluation indicator of ranking accuracy, as defined in Eq. (7).

$$Recall_{\mathrm{top}-N} = \frac{TP}{TP + FN}, \tag{7}$$

where TP and FN represent the number of correctly recommended top-N algorithms and incorrectly recommended top-N algorithms, respectively; the sum of TP and FN is N. The $Recall$ focuses on whether the top-N algorithms in the true ranking are consistent with the top-N regression algorithms in the predicted ranking.

3. Results and discussion

3.1. Experimental setup

In order to evaluate the robustness and general applicability of our proposed AutoML system on a broad range of datasets, the performance of Auto-MatRegressor is validated by using the leave-one-out procedure. At each step, 53 materials datasets are used as the training set, and the remaining dataset is used to test Auto-MatRegressor. For each dataset, it is further randomly divided into training set (80% of samples) for model construction and test set (20% of samples) for performance evaluation, respectively. Furthermore, the same pre-processing is applied to all datasets: removing missing values, one-hot encoding for categorical features, and standardizing all features. The meta-features are also standardized in Auto-MatRegressor to eliminate the dimensional influence between different meta-features. This procedure is repeated 100 times, where the number of similar materials datasets and similar general datasets is set at 9 and 3, respectively. The SRC on 18 regression algorithms and the $Recall$ of the top three regression algorithms are considered for the evaluation of the recommendation accuracy. Furthermore, six extended validation

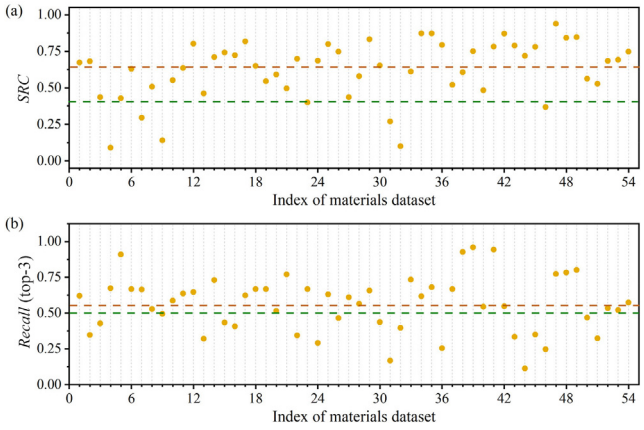

Fig. 4. (Color online) The average ranking accuracy of 100 rounds of experiments. (a) Average SRC values for 18 regression algorithms on different materials datasets. (b) Average $Recall$ values of the recommended top-3 regression algorithms on different materials datasets. The green dotted line represents the standard threshold for comparison, and the orange dotted line represents the mean of SRC values or $Recall$ values of all 54 datasets.

datasets are adopted to further verify the generalization and robustness of Auto-MatRegressor on structured datasets.

3.2. Recommendation ability of Auto-MatRegressor

The *SRC* value on each analysed dataset is shown in Fig. 4a. Around 60% of the datasets achieve greater accuracy than the mean *SRC* of 0.620. Especially for materials datasets 17, 29, 34, 35, 42, 47, 48, and 49, their *SRC* are 31.9%, 34.3%, 40.8%, 40.9%, 40.5%, 51.5%, 35.9%, and 36.6% higher than the mean value, respectively. From Fig. 4b, we can also observe that among the datasets whose *SRC* values are not very outstanding, such as datasets 4, 7, and 23, the *Recall* values of the top-3 algorithms are greater than the mean *Recall* of 0.560. The results indicate that Auto-MatRegressor provides an effective and efficient algorithm recommendation strategy for most experimental datasets. In addition, according to the table of critical values of *SRC*, at a significance level of 5% (one-sided test), Auto-MatRegressor achieves a significantly higher experimental *SRC* than the threshold of 0.401 on 87% of datasets.

For the end user, the process of constructing a ML model with Auto-MatRegressor consists of three steps, i.e., meta-features computation, collaborative recommendation, and further hyperparameter optimization. The first two steps serve to recommend the most suitable regression algorithms. Compared with the classic CASH method, it can automatically complete algorithm selection in shorter runtime from a few seconds to minutes (as shown in Table S7 online). Note that for the datasets with suboptimal *Recall* values of the top three regression algorithms, Auto-MatRegressor can find the true top algorithms in far less time than optimizing all algorithms from scratch. Therefore, it is demonstrated that Auto-MatRegressor can recommend promising regressors for different materials datasets more quickly and efficiently.

3.3. Property prediction by the promising regressors

To evaluate the prediction performance of the algorithms identified by Auto-MatRegressor, 12 materials datasets are chosen for analysis, as shown in Table 2. It can be seen that these datasets cover various numbers of features, with the number of samples ranging from 20 to 7230. In addition, the datasets describe different target properties of diverse materials, including the density of biofuel compounds, the superconducting transition temperature of Fe-based superconductors, the oxide ionic conductivity of perovskite materials, etc. Noteworthy, we not only consider datasets with high recommendation accuracy but also use inferior datasets to comprehensively validate the predictive ability of Auto-MatRegressor: the 5 datasets with both high *SRC* and high *Recall*; the 5 datasets with high *SRC* and low *Recall*; the other datasets with low *SRC* and low *Recall*. For each experimental dataset, the average predicted rankings of 18 regression algorithms in 100 experiments on different training sets are implemented. Then, the most promising regressors recommended by Auto-MatRegressor are used to predict the target properties of the remaining test samples.

As shown in Fig. 5, the rankings of multiple linear models such as LR, Lasso, Ridge, LS, and RANSAC are not very high. On the contrary, non-linear models, such as KNN, AdaBoost, MLP, KRR, and RF have higher predicted rankings than other models. It is attributed to the fact that the complex physicochemical mechanisms of the materials properties cause linear models to fail in describing the nonlinear relationships between various material features and target properties.

The prediction accuracy of the automatically recommended regressors is shown in Fig. 6. The prediction accuracy reported in the original research article for each dataset was used as the baseline for comparison. The results show that the models AdaBoost, MLP, SGD, and KRR have excellent prediction performance, achieving lower average RMSE values in most cases. Moreover, at least one of the models recommended by Auto-MatRegressor achieved better prediction accuracy than the baseline method. For MD5, MD21, and MD27, the RMSEs of the recommended regression models were slightly lower but comparable to those in the source articles. This was because we used a less costly hyperparameter search space than exhausting all hyperparameters to achieve a shorter modelling time. The superior results effectively demonstrated the good generalization and applicability of the Auto-MatRegressor to recommend regressors from the perspective of materials datasets with varying data sizes and target property tasks. Considering the trade-off between model complexity and time efficiency of constructing meta-data, the optimization space of hyperparameters is appropriately reduced in this paper. Additionally, PAR, SGD, and ElaNet also perform well on MD20, MD28, MD45, MD52, and MD54. It indicates that not only the commonly used models but also other less commonly used models should be considered in properties prediction research since they potentially can achieve higher prediction accuracy, and Auto-MatRegressor allows doing that.

3.4. Application examples

As described in Section 2.1, Auto-MatRegressor only requires the user to input the dataset at the beginning and is then able to automatically complete the model construction and finally make predictions for target properties in a short time. To further illustrate the effectiveness and reliability of Auto-MatRegressor, three materials datasets, namely MD29, MD43, and MD44 (detailed in

Table 2
The list of selected materials datasets.

Dataset	Material	TP[a]	N[b]	F[b]	Source
MD5	Biofuel compound	Density	5619	47	[48]
MD13	Fe-based superconductor	Superconducting transition temperature	33	2	[49]
MD14	Methane hydrate	Formation temperature	702	6	[13]
MD20	High entropy alloy	Hardness	155	6	[23]
MD21	Metallic host	Activation energy of solute diffusion	408	11	[17]
MD22	ABO_3 perovskite	Oxide ionic conductivity	128	6	[50]
MD27	Ni-based single crystal superalloy	Creep rupture life	266	27	[34]
MD28	Superconducting doped MgB_2	Superconducting transition temperature	20	2	[51]
MD32	ABO_3 perovskite	Conductivity	7230	112	[35]
MD45	SiO_2-based glass	Shear	498	511	[16]
MD52	Waste-based material	Compressive strength	207	4	[52]
MD54	Waste-based material	Pore density	207	4	[52]

[a] TP is the target property of materials datasets.
[b] N and F are the number of samples and the number of materials descriptors, respectively.

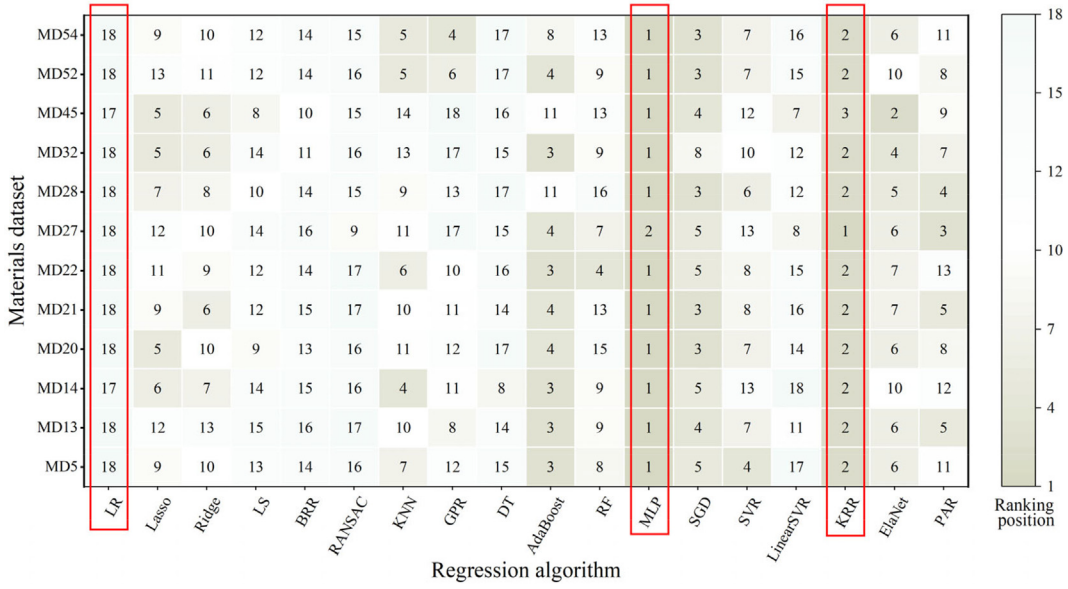

Fig. 5. (Color online) The average rankings of all 18 regression methods in 100 rounds of experiments. The label of the cell in the heatmap indicates the ranking position, i.e., "1" represents the excellent ranking; "18" represents the lowest ranking. The deeper the color, the higher the ranking position.

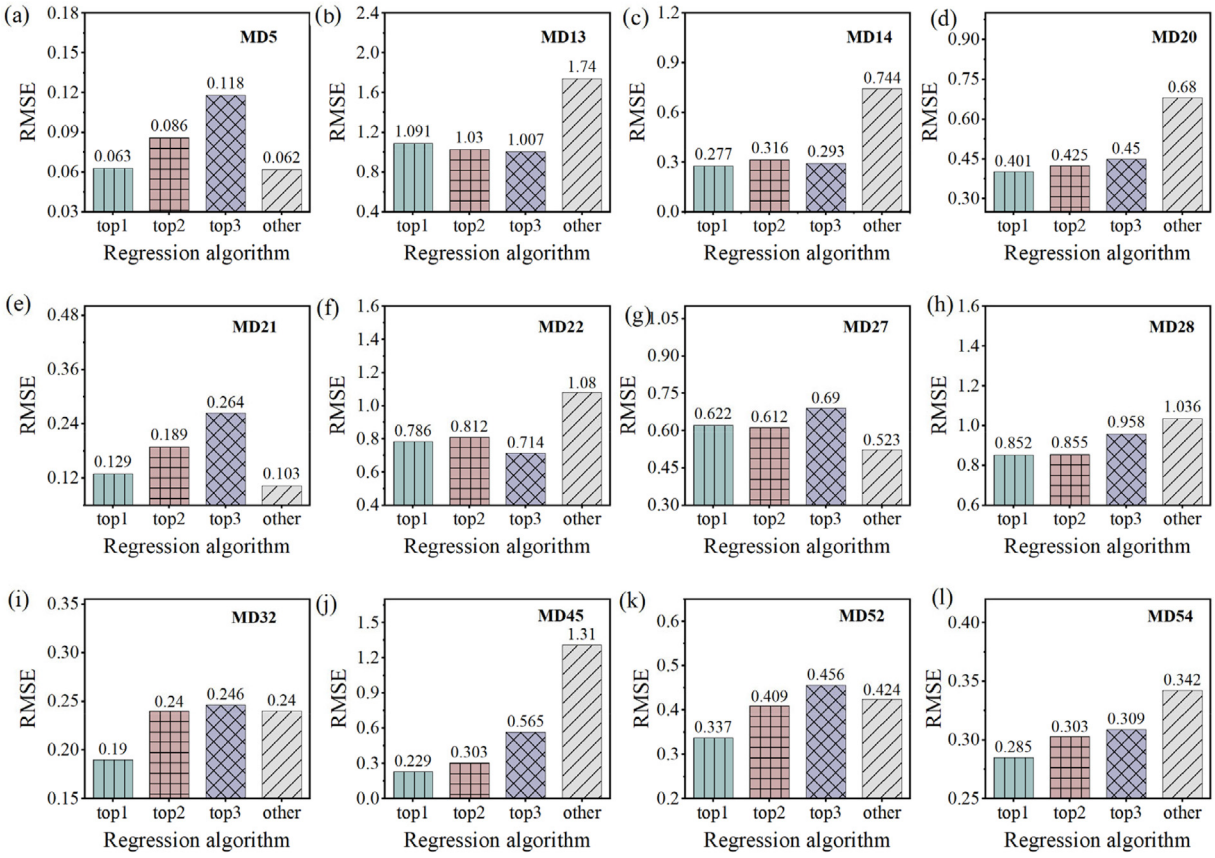

Fig. 6. (Color online) Prediction performance of recommended algorithms on different datasets. (a) Density prediction on MD5. (b) Superconducting transition temperature prediction on MD13. (c) Formation temperature prediction on MD14. (d) Hardness prediction on MD20. (e) Activation energy of solute diffusion prediction on MD21. (f) Oxide ionic conductivity prediction on MD22. (g) Creep rupture life prediction on MD27. (h) Superconducting transition temperature prediction on MD28. (i) Conductivity prediction on MD32. (j) Shear prediction on MD45. (k) Compressive strength prediction on MD52. (l) Pore density prediction on MD54. "top1", "top2", and "top3" represent the top three regression models recommended by Auto-MatRegressor, respectively; "other" represents the baseline model in the original research article. The shorter the length of the histogram, the smaller the RMSE value, that is, the higher the prediction accuracy.

Table S1 online), were selected in consideration of the comparability between the prediction results realized by Auto-MatRegressor and those in the original articles. For each dataset, to avoid model over-fitting, all samples were randomly divided into the 80% training set and 20% test set, and the recommended models were evaluated by the average RMSE on 5-fold CV. The results of the

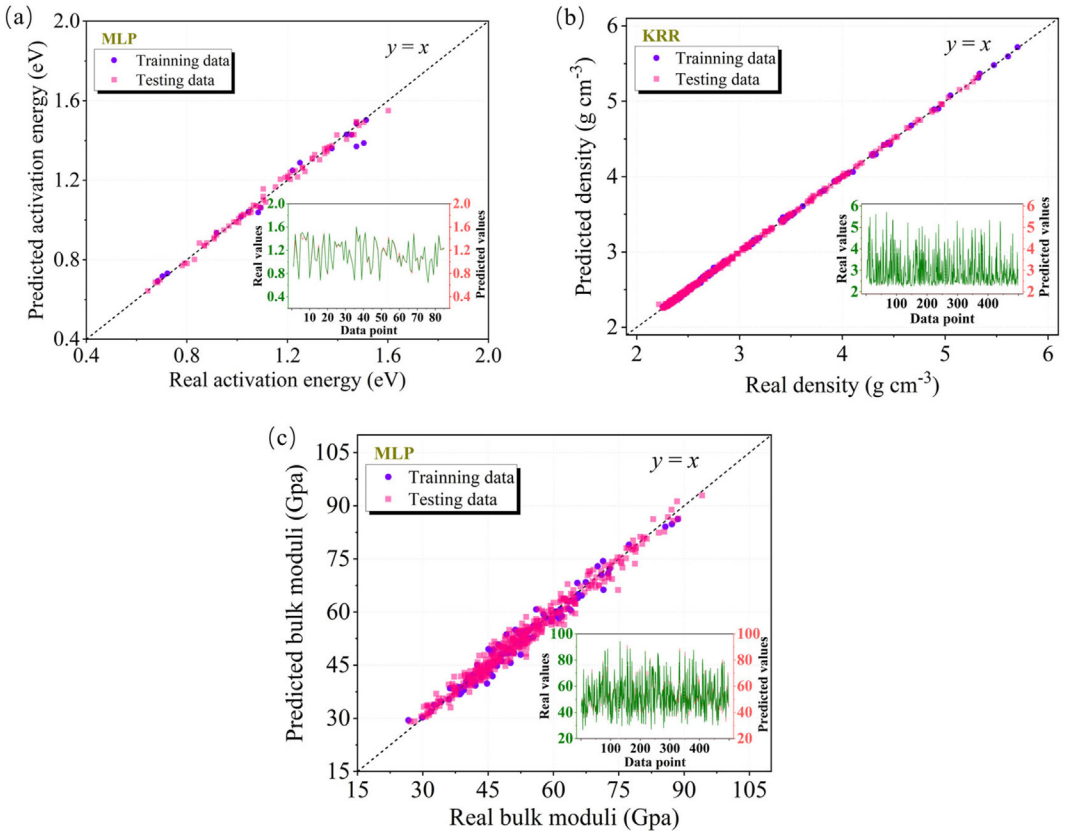

Fig. 7. (Color online) Predictions of target properties of different materials datasets. Scatter plots showing true values versus predicted values via recommended top-one ML model by Auto-MatRegressor. (a) Activation energy prediction of NASICON-type solid electrolytes (materials dataset 29). (b) Density prediction of SiO_2-based glass (materials dataset 43). (c) Bulk moduli prediction of SiO_2-based glass (materials dataset 44). The line $y = x$ is plotted to show the deviation from perfect predictions, and subplots in the lower right-hand corner are used to show in detail the real value and the predicted value of each data point.

recommended regression models are visualized in Fig. 7 with true values on the horizontal axis and predicted values on the vertical axis. As can be seen, the scatter points are close to the diagonal line indicating a good ability to capture true information.

For activation energy prediction of NASICON material, Auto-MatRegressor can automatically achieve model construction by learning from previous modeling experience on similar activation energy prediction tasks of NASICON-type solid electrolytes. As shown in Fig. 7a, MLP performs well with an RMSE of 0.045 eV. It is particularly close to the baseline RMSE achieved in the original research article that focused on applying multiple ML models to verify the proposed descriptor recognizer [53]. Furthermore, it can be observed from Fig. 7b, c that KRR and MLP were preferentially recommended for predicting the density and bulk moduli of SiO_2-based glass material [16]. The regression RMSEs were 0.017 g cm^{-3} and 2.120 GPa, which were improved by 25.7% and 29.1% than those in their original articles, respectively. The results demonstrate that Auto-MatRegressor can effectively eliminate the need for materials researchers to carry out laborious and time-consuming experiments for constructing ML models with high accuracy.

3.5. Ablation study

To further validate Auto-MatRegressor, the ablation experiments were carried out. As can be seen from Table 3, compared with the models using a single type of meta-data, our proposed model always achieves a higher *SRC* and a higher *Recall*, which is not surprising, since modeling needs much data training to achieve

Table 3
Ablation experiment for Auto-MatRegressor.

	Collaborative mechanism		MCT-DK	Traditional meta-features		Traditional and three enhanced meta-features	
	MMD[a]	GMD[a]		*SRC* [b]	*Recall* [b]	*SRC* [b]	*Recall* [b]
Auto-MatRegressor	√			**0.5312**	**0.5256**	0.5459	0.5299
		√		0.4395	0.4745	0.4734	0.5135
	√	√		0.5888	0.5369	0.6082	0.5573
	√		√	0.5490	0.5305	0.5587	0.5325
	√	√	√	0.6001	0.5374	**0.6169**	**0.5599**

[a] "MMD" and "GMD" represent the abbreviations of "meta-data based on materials datasets" and "meta-data based on general datasets", respectively.
[b] "*SRC*" and "*Recall*" are the mean of Spearman's rank correlation values on all algorithms and the mean of *Recall* values on the recommended top-3 algorithms in 100 runs of experiments, respectively. The symbol of "√" represents whether a strategy is considered for introduction into the original meta-learning recommendation process.

higher prediction accuracy. Furthermore, employing meta-learning based on materials meta-data allows for achieving better predictive results than that based on general meta-data. It indicates that modeling experience from similar materials tasks is more informative than that from other unrelated tasks. The models embedded with MCT-DK also have higher recommendation accuracy than those without MCT-DK, which is attributed to the fact that materials domain knowledge strongly improves the ranking of each regression algorithm on similar datasets. Namely, the more similar the materials category of the previous tasks is to that of the given task, the more preferentially the modeling experience on similar tasks will be introduced.

The introduction of enhanced meta-features also enables the model to obtain significantly higher recommendation accuracy. It can be concluded that the enhanced meta-features characterizing the data distribution and uncertainty of the target properties can better distinguish between materials datasets and general datasets. Furthermore, compared with the traditional meta-learning approach (i.e., without general meta-data, enhanced features, and guidance of materials domain knowledge), Auto-MatRegressor proposed in this paper can obtain higher *SRC* and *Recall* with an average significant increase of 16.1% and 6.5%, respectively. In conclusion, these results well proved the effectiveness and reliability of our proposed approach.

3.6. Generalization verification

To further verify the generalization and robustness of the Auto-MatRegressor, six newly reported materials datasets from 2021 to 2022 are collected, as shown in Table 4 (detailed in Table S3 online). The performance of the most promising ML models con-

Table 4
List of 6 newly reported materials datasets.

Dataset	Material	TP[a]	N^b	F^b	Source
val_1	Solid-state electrolyte	Ionic conductivity	176	14	[54]
val_2	Metal oxide material	Reduction temperature	38	20	[55]
val_3	Perovskite oxides	Bandgap	1016	81	[56]
val_4	*Ni*-based crystal superalloy	Creep rupture life	393	96	[57]
val_5	*Ni*-based crystal superalloy	Creep rupture life	36	96	[57]
val_6	Bulk metallic glasses	Glass transition temperature	54	7	[58]

[a] TP is the target property dataset.
[b] *N* and *F* are the number of samples and the number of materials descriptors, respectively.

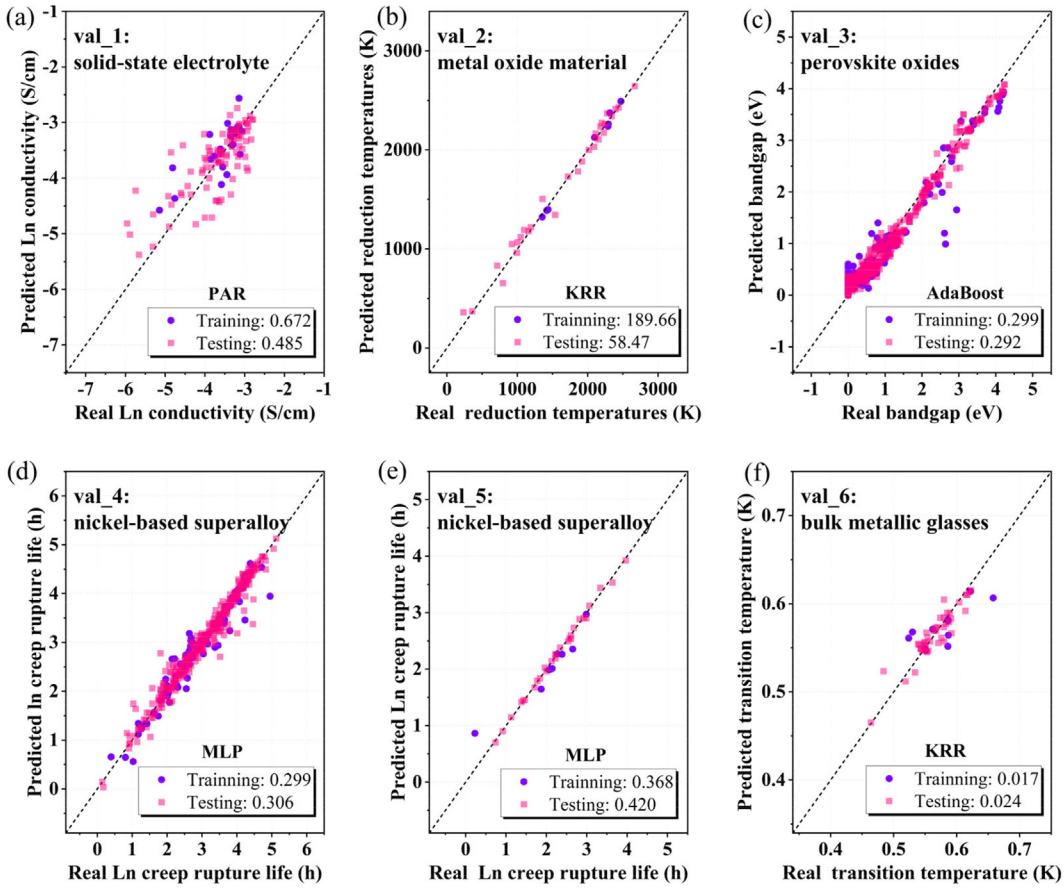

Fig. 8. (Color online) Prediction results of target properties of extended materials datasets. (a) Ionic conductivity prediction on dataset val_1. (b) Reduction temperature prediction on dataset val_2. (c) Bandgap prediction on dataset val_3. (d) Creep rupture life prediction on dataset val_4. (e) Creep rupture life prediction on dataset val_5. (f) Glass transition temperature prediction on dataset val_6. Scatter plots show true values versus predicted values via recommended top-one ML model by Auto-MatRegressor. The values in each subgraph represent the RMSEs respectively on the training set and test set obtained by the most promising ML model.

ducted on the extended datasets is shown in Fig. 8. The RMSEs of the datasets val_2, val_3, and val_6 show that compared with the source literature, Auto-MatRegressor has a comparable or better predictive ability. For val_1, the ionic conductivity of solid-state electrolyte material is predicted by Auto-MatRegressor, and then a threshold mentioned in Ref. [54] is used for classifying them, where the classification accuracy is 0.818 comparable with that obtained in the source literature. For val_4 and val_5, MAPE was added as the evaluation indicator (detailed in the Supplementary materials) and the accuracy of Auto-MatRegressor achieved 85.9% and 8.1% improvement compared with those in the source literature. Above all, experimental results show that using Auto-MatRegressor to recommend ML algorithms can reduce the modeling time and perform accurate predictions for unknown materials tasks.

4. Conclusion

Just like alchemists, materials researchers often use a "trial-and-error" approach to select an appropriate regression algorithm and optimize its hyperparameters. AutoML aims to automate the modeling process to reduce user intervention; however, exhaustive optimization of a huge hyperparameter space is often computationally expensive. To overcome the shortcomings of traditional ML modeling, meta-learning is introduced to improve existing AutoML technologies. Meta-learning can be seen as a subfield of AutoML, which allows new tasks to learn from modeling experience of prior similar tasks to enable modeling hot-starting rather than starting training from scratch. In general, a relatively large and ideally experimental database is essential for meta-learning-based AutoML to succeed. However, a limited number of available materials datasets means insufficient meta-data, thereby easily resulting in the overfitting problem. Furthermore, due to little research on meta-learning-based AutoML specifically in the field of materials property prediction, the existing general meta-features may not be sufficient to characterize new datasets. In addition, pure data-driven ML models are often "black boxes", and the training relies entirely on empiricism and statistical characteristics of materials data. This may be a crucial reason that leads to the phenomenon that the learning results are often inconsistent or even contradictory to domain knowledge.

To improve the situation, Auto-MatRegressor based on meta-learning is proposed to automate the modeling process for materials property prediction. In this work, we establish a comprehensive experimental database, which consists of 54 materials datasets from literature and 60 general datasets from online sources. Since the datasets have been publicly released, they are expected to be highly reliable. Concurrently, the regression-oriented meta-data is constructed based on 18 regression algorithms commonly used in materials science and 27 meta-features. Therein, meta-features are comprised of 24 traditional meta-features as well as 3 newly proposed enhanced meta-features from the perspectives of data distribution and uncertainty of target attributes. The experiments demonstrated that these meta-features can characterize the dataset more comprehensively and enhance meta-learning for algorithm recommendation. Auto-MatRegressor also employs collaborative meta-learning to balance different recommended results, where the MCT-DK is designed to support the encoding and embedding of domain expert knowledge into the modeling process. It is suggested that the ratio of the number of materials datasets to the number of general datasets and the diversity of materials categories are crucial factors influencing the recommendation results.

Auto-MatRegressor aims to solve the shortcomings of traditional ML modeling for structured datasets. In the future, with the increase of high-quality data volume and computing resources, deep learning models are expected to be integrated into Auto-MatRegressor for larger datasets. Additionally, the flourishing of the data representation and data quality and quantity governance technologies can enable the multi-source and heterogeneous materials data in public databases (e.g., MP, OQMD, AFLOWlib) to be employed directly in ML model training (e.g., General Pre-Training model). We believe that Auto-MatRegressor has a high potential for the accurate prediction of materials properties, thus accelerating materials discovery and design.

Conflict of interest

The authors declare that they have no conflict of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (52073169 and 92270124), the National Key Research and Development Program of China (2021YFB3802100), and the Key Research Project of Zhejiang Laboratory (2021PE0AC02). We appreciated the High Performance Computing Center of Shanghai University and Shanghai Engineering Research Center of Intelligent Computing System for providing the computing resources and technical support.

Author contributions

Yue Liu contributed to the conceptualization, methodology, validation, writing, editing, and supervision. Shuangyan Wang contributed to the methodology, software, validation, data curation, writing, and editing. Zhengwei Yang contributed to the conceptualization, methodology, writing, and editing. Maxim Avdeev contributed to the writing and editing. Siqi Shi contributed to the conceptualization, methodology, validation, writing, and supervision. All authors participate in discussing the results and comments on the manuscript.

Data availability

The authors declare that the main data supporting the finding of this study are available within the article and its Supplementary materials files. Extra data and source codes for the experiments are available from the corresponding author upon reasonable request. The authors also developed an online tool for Auto-MatRegressor, which is integrated into the Electrochemical Energy Storage Materials Design Platform (https://matgen.nscc-gz.cn/solidElectrolyte/machine_learning/).

Appendix A. Supplementary materials

Supplementary materials to this article can be found online at https://doi.org/10.1016/j.scib.2023.05.017.

References

[1] Liu Y, Zhao T, Ju W, et al. Materials discovery and design using machine learning. J Materiomics 2017;3:159–77.
[2] Liu Y, Guo B, Zou X, et al. Machine learning assisted materials design and discovery for rechargeable batteries. Energy Storage Mater 2020;31:434–50.
[3] Wang A-Y-T, Murdock RJ, Kauwe SK, et al. Machine learning for materials scientists: an introductory guide toward best practices. Chem Mater 2020;32:4954–65.
[4] Jain A, Ong SP, Hautier G, et al. Commentary: the materials project: a materials genome approach to accelerating materials innovation. APL Mater 2013;1:011002.
[5] Curtarolo S, Setyawan W, Hart GLW, et al. AFLOW: an automatic framework for high-throughput materials discovery. Comput Mater Sci 2012;58:218–26.

[6] Saal JE, Kirklin S, Aykol M, et al. Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD). JOM 2013;65:1501–9.
[7] Dunn A, Wang Q, Ganose A, et al. Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm. npj Comput Mater 2020;6:138.
[8] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1:67–82.
[9] Chen C, Zuo Y, Ye W, et al. A critical review of machine learning of energy materials. Adv Energy Mater 2020;10:1903242.
[10] Kaufmann K, Maryanovsky D, Mellor WM, et al. Discovery of high-entropy ceramics via machine learning. npj Comput Mater 2020;6:42.
[11] Wen C, Wang C, Zhang Y, et al. Modeling solid solution strengthening in high entropy alloys using machine learning. Acta Mater 2021;212:116917.
[12] Roman D, Saxena S, Robu V, et al. Machine learning pipeline for battery state-of-health estimation. Nat Mach Intell 2021;3:447–56.
[13] Xu H, Jiao Z, Zhang Z, et al. Prediction of methane hydrate formation conditions in salt water using machine learning algorithms. Comput Chem Eng 2021;151:107358.
[14] Fabian P, Gaël V, Alexandre G, et al. Scikit-learn: machine learning in python. J Mach Learn Res 2011;12:2825–30.
[15] Wang Z, Sun Z, Yin H, et al. Data-driven materials innovation and applications. Adv Mater 2022;34:2104113.
[16] Hu Y-J, Zhao G, Zhang M, et al. Predicting densities and elastic moduli of SiO_2-based glasses by machine learning. npj Comput Mater 2020;6:25.
[17] He K, Kong X, Liu CS. Robust activation energy predictions of solute diffusion from machine learning method. Comput Mater Sci 2020;184:109948.
[18] Colorni A, Dorigo M, Maniezzo V, et al. Distributed optimization by ant colonies. Proceedings of the European conference on artificial life (ECAL) 1992;142:134–42.
[19] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the International Conference on Neural Networks (ICNN) 1995;4:1942–8.
[20] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6:182–97.
[21] Shields BJ, Stevens J, Li J, et al. Bayesian reaction optimization as a tool for chemical synthesis. Nature 2021;590:89–96.
[22] Ozaki Y, Suzuki Y, Hawai T, et al. Automated crystal structure analysis based on blackbox optimisation. npj Comput Mater 2020;6:75.
[23] Wen C, Zhang Y, Wang C, et al. Machine learning assisted design of high entropy alloys with desired property. Acta Mater 2019;170:109–17.
[24] Kotthoff L, Thornton C, Hoos HH, et al. Auto-WEKA 2.0: automatic model selection and hyperparameter optimization in WEKA. J Mach Learn Res 2016;17:1–5.
[25] Olson RS, Moore JH. TPOT: a tree-based pipeline optimization tool for automating machine learning. Proc Mach Learning Res 2016;64:66–74.
[26] Komer B, Bergstra J, Eliasmith C. Hyperopt-Sklearn: automatic hyperparameter configuration for scikit-Learn. Proceedings of the Python in Science Conference (SciPy) 2014;1:32–7.
[27] Pimentel BA, de Carvalho ACPLF. A new data characterization for selecting clustering algorithms using meta-learning. Inf Sci 2019;477:203–19.
[28] Agarwal M, Yurochkin M, Sun Y. On sensitivity of meta-learning to support data. Proceedings of the Neural Information Processing Systems (NeurIPS) 2021;34:20447–60.
[29] Qiu YL, Zheng H, Devos A, et al. A meta-learning approach for genomic survival analysis. Nat Commun 2020;11:6350.
[30] Matthias F, Aaron K, Katharina E, et al. Efficient and robust automated machine learning. Proceedings of the Neural Information Processing Systems (NeurIPS) 2015;28:2962–70.
[31] Pimentel BA, de Carvalho ACPLF. A Meta-learning approach for recommending the number of clusters for clustering algorithms. Knowl Based Syst 2020;195:105682.
[32] Aguiar GJ, Santana EJ, de Carvalho ACPFL, et al. Using meta-learning for multi-target regression. Inf Sci 2022;584:665–84.
[33] Hospedales T, Antoniou A, Micaelli P, et al. Meta-learning in neural networks: a survey. IEEE Trans Pattern Anal Mach Intell 2022;44:5149–69.
[34] Liu Y, Wu J, Wang Z, et al. Predicting creep rupture life of Ni-based single crystal superalloys using divide-and-conquer approach based machine learning. Acta Mater 2020;195:454–67.
[35] Priya P, Aluru NR. Accelerated design and discovery of perovskites with high conductivity for energy applications through machine learning. npj Comput Mater 2021;7:90.
[36] Fulkerson B. Machine learning, neural and statistical classification. Technometrics 1995;37:459.
[37] Rivolli A, Garcia LPF, Soares C, et al. Meta-features for meta-learning. Knowl Based Syst 2022;240:108101.
[38] Liu Y, Niu C, Wang Z, et al. Machine learning in materials genome initiative: a review. J Mater Sci Technol 2020;57:113–22.
[39] Pilania G, Gubernatis JE, Lookman T. Multi-fidelity machine learning models for accurate bandgap predictions of solids. Comput Mater Sci 2017;129:156–63.
[40] Jia X, Lynch A, Huang Y, et al. Anthropogenic biases in chemical reaction data hinder exploratory inorganic synthesis. Nature 2019;573:251–5.
[41] Doan HA, Agarwal G, Qian H, et al. Quantum chemistry-informed active learning to accelerate the design and discovery of sustainable energy storage materials. Chem Mater 2020;32:6338–46.
[42] Lei B, Kirk TQ, Bhattacharya A, et al. Bayesian optimization with adaptive surrogate models for automated experimental design. npj Comput Mater 2021;7:194.
[43] Liu Y, Zou X, Ma S, et al. Feature selection method reducing correlations among features by embedding domain knowledge. Acta Mater 2022;238:118195.
[44] Liu Y, Zou X, Yang Z, et al. Machine learning embedded with materials domain knowledge. J Chin Cera Soc 2022;50:863–76.
[45] de Winter J, Gosling S, Potter J. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: a tutorial using simulations and empirical data. Psychol Methods 2016;21:273–90.
[46] Ferrari DG, de Castro LN. Clustering algorithm selection by meta-learning systems: a new distance-based problem characterization and ranking combination methods. Inf Sci 2015;301:181–94.
[47] Cunha T, Soares C, de Carvalho ACPLF. Metalearning and recommender systems: a literature review and empirical study on the algorithm selection problem for collaborative filtering. Inf Sci 2018;423:128–44.
[48] Saldana DA, Starck L, Mougin P, et al. Prediction of density and viscosity of biofuel compounds using machine learning methods. Energy Fuels 2012;26:2416–26.
[49] Owolabi TO, Akande KO, Olatunji SO. Prediction of superconducting transition temperatures for Fe-based superconductors using support vector machine. Adv Phys 2014;35:12–26.
[50] Liu X, Lu W, Peng C, et al. Two semi-empirical approaches for the prediction of oxide ionic conductivities in ABO_3 perovskites. Comput Mater Sci 2009;46:860–8.
[51] Owolabi TO, Akande KO, Olatunji SO. Estimation of superconducting transition temperature TC for superconductors of the doped MgB_2 system from the crystal lattice parameters using support vector regression. J Supercond Nov Magn 2015;28:75–81.
[52] Anghelescu L, Cruceru M, Diaconu B. Building materials obtained by recycling coal ash and waste drilling fluid and characterization of engineering properties by means of artificial neural networks. Constr Build Mater 2019;227:116616.
[53] Liu Y, Ge X, Yang Z, et al. An automatic descriptors recognizer customized for materials science literature. J Power Sources 2022;545:231946.
[54] Adhyatma A, Xu Y, Hawari NH, et al. Improving ionic conductivity of doped $Li_7La_3Zr_2O_{12}$ using optimized machine learning with simplistic descriptors. Mater Lett 2022;308:131159.
[55] Garrido Torres JA, Gharakhanyan V, Artrith N, et al. Augmenting zero-Kelvin quantum mechanics with machine learning for the prediction of chemical reactions at high temperatures. Nat Commun 2021;12:7012.
[56] Ihalage A, Hao Y. Analogical discovery of disordered perovskite oxides by crystal structure information hidden in unsupervised material fingerprints. npj Comput Mater 2021;7:75.
[57] Zhu Y, Duan F, Yong W, et al. Creep rupture life prediction of nickel-based superalloys based on data fusion. Comput Mater Sci 2022;211:111560.
[58] Chang D, Lu W, Wang G. Designing bulk metallic glasses materials with higher reduced glass transition temperature via machine learning. Chemometr Intell Lab 2022;228:104621.

Yue Liu obtained her B.S. and M.S. degrees in Computer Sciences from Jiangxi Normal University in 1997 and 2000, respectively. She got her Ph.D. degree in Control Theory and Control Engineering from Shanghai University (SHU) in 2005. She was a curriculum R&D manager at the Sybase-SHU IT Institute of Sybase Inc. from July 2003 to July 2004 and a visiting scholar at the University of Melbourne from September 2012 to September 2013. At present, she is a professor at SHU. Her current research interest focuses on the research of machine learning, data mining, and AI for materials science.

Siqi Shi obtained his B.S. and M.S. degrees from Jiangxi Normal University in 1998 and in 2001, respectively. He got his Ph.D. degree from Institute of Physics, Chinese Academy of Sciences in 2004. After that, he joined the National Institute of Advanced Industrial Science and Technology, Japan and Brown University, USA as a senior research associate, respectively. In early 2013, he joined Shanghai University as a professor. His current research interest focuses on the fundamentals and multiscale calculation of electrochemical energy storage materials and materials design and performance optimization using machine learning.