COMMUNICATIONS MATERIALS

ARTICLE

https://doi.org/10.1038/s43246-020-00052-8 OPEN

Integrating multiple materials science projects in a single neural network

Kan Hatakeyama-Sato[1] & Kenichi Oyaizu[1]✉

In data-intensive science, machine learning plays a critical role in processing big data. However, the potential of machine learning has been limited in the field of materials science because of the difficulty in treating complex real-world information as a digital language. Here, we propose to use graph-shaped databases with a common format to describe almost any materials science experimental data digitally, including chemical structures, processes, properties, and natural languages. The graphs can express real world's data with little information loss. In our approach, a single neural network treats the versatile materials science data collected from over ten projects, whereas traditional approaches require individual models to be prepared to process each individual database and property. The multitask learning of miscellaneous factors increases the prediction accuracy of parameters synergistically by acquiring broad knowledge in the field. The integration is beneficial for developing general prediction models and for solving inverse problems in materials science.

[1] Department of Applied Chemistry, Waseda University, Tokyo 169-8555, Japan. ✉email: oyaizu@waseda.jp

Data-driven science is becoming increasingly important amidst the worldwide deluge of data[1,2]. Recent developments in deep learning have provided a way to extract important features from big data automatically and to understand new phenomena[3]. Integration of data by machine learning is also important in materials science. New devices, such as next-generation batteries and photovoltaics, could be developed more efficiently by automatically exploring materials with superior properties, chemical structures, and processes[4,5].

Despite the high expectations around materials informatics, even cutting-edge prediction models are not yet able to integrate big data from materials science, due to the lack of general knowledge in this field[6–13]. A number of models predict a variety of material parameters, including physical and chemical properties, structures, and spectroscopic responses[6–18]. The recent development of data mining techniques from scientific literature is also helpful to increase the number of databases and to enhance prediction accuracy[14,15,18]. However, a critical drawback has been that the previous models could not predict more than two parameters (Table 1 and Fig. 1a, b) and contained as many individual prediction algorithms and models as predicting parameters[6–13]. Therefore, the models could not perform essential tasks that are easy for humans, such as learning, considering, and predicting multiple real-world phenomena with a single intelligence. This limitation arises from the use of traditional, inflexible table databases. To integrate knowledge, varied information must be inputted and outputted (i.e., multimodal learning)[19,20].

In this study, we introduced graphs with a common format to integrate diverse materials science projects (Fig. 2a, Supplementary Figs. 1 and 2). The format can express almost all experimental materials science information, such as structure, properties, processes, text, images, and even sounds. All related information from more than ten projects was inputted into a single neural network to predict more than 40 parameters simultaneously, including numeric properties, chemical structures, and text (Fig. 1). Graph approaches have been employed to analyze the relationships of atom-connections, chemical features, and reactions[6,16,17]. In this study, we extended the approach to train a neural network with the general phenomena of science, which are expressed by graphs. The multitask training of versatile information was essential to acquire broad knowledge about materials science. Our graph approach will be the key to developing general-purpose artificial intelligence for materials science, including inverse problem solving.

Results

Process informatics for electron-conducting polymers. As a model case to demonstrate the effect of the graph format, we examined the process informatics of poly(3,4-ethylenedioxythiophene) doped with poly(4-styrenesulfonate) (PEDOT-PSS; Fig. 3). The polymer is known for its high electron conductivity and can be used in transparent flexible conductive films, capacitors, solar cells, thermoelectrics, and other energy-related devices[21,22]. The conductivity reaches over 3000 S cm^{-1} after the careful chemical treatment of the polymer film[22,23]. To achieve higher conductivity, a number of new annealing methods, including repeated chemical treatments with strong acids, bases, and solvents, have been reported[23]. Informatic approaches have been partially introduced to predict the properties of PEDOT-PSS[24,25]. However, the post-treatment methods have become too long and complex to be analyzed by conventional machine learning approaches or to be understood except by a few specialists (example scheme is shown in Fig. 3a).

Process informatics aims to optimize procedures by using statistical tools[9,26,27]. The supposedly important factors for the target performance are extracted manually, and recorded in table databases (e.g., heating temperature, mixing speed, and duration; Supplementary Fig. 1). The table format is normally used because most machine learning models can only accept numeric arrays[3].

The intrinsic problems with the traditional approach are the inflexible format of the table and ignorance of the experimental context. The database format must be changed whenever a new experimental step (e.g., additional mixing) is considered, although additional steps are often examined to optimize the procedure. It may not be possible to describe complex experimental information fully in a numeric table alone. Even if the table is constructed,

Table 1 Comparison of the present work with previous prediction models.

	Targets	Predictable parameters[a]	Inputted databases[b]	Input[b]			Output[b]			Ref.
				Num.	Chem.	Misc.	Num.	Chem.	Misc.	
1	Properties, chemicals, processes, etc.	>40	>10	+	+	+	+	+	+	This work
2	Ionic conductivity	1	1 (2)[c]	+	+		+			6
3	Material properties	1 (>40)[c]	1 (>10)[c]		+		+			8
4	Polymer properties	1 (7)[c]	1		+		+			7
5	Nanosheet yield	1	1		+		+			9
6	OLED performance[d]	1	1		+		+			11
7	Chemical reaction[e]	1	1		+		+			12
8	NMR spectra[f]	1	1			+			+	10
9	Molecular structure[g]	1	1		+			+		13
10	Inorganic semiconductors, etc.[h]	-	(1)			+			+	18
11	Properties, chemicals, etc.[i]	(>3)	(>3)	(+)	(+)	(+)	(+)	(+)	(+)	17

[a] Number of parameters and databases predicted or interpreted by a single prediction model.
[b] Inputs and outputs of the models. "Num.", "Chem.", and "Misc." indicate numbers, chemical structures, and miscellaneous parameters (e.g., images, sounds, and spectra). For example, prediction model number 2 in the table predicts ionic conductivity (numbers) of polymer composites from their chemical structures and numeric parameters (composition ratio, etc).
[c] Multiple models were generated to input/predict multiple databases and parameters by transfer learning. Each model shares mutual algorithms for the improved recognition of important features, but final predictions themselves are done individually (see Supplementary Fig. 12).
[d] OLED: organic light emitting diode.
[e] Predict yield of chemical reactions.
[f] Predict NMR spectra of chemicals.
[g] Known as an autoencoder of chemical structures.
[h] From text abstracts, the model calculates the embedding vectors of chemicals, expressed by words.
[i] Individual databases and machine learning models were connected and analyzed by graph approaches.

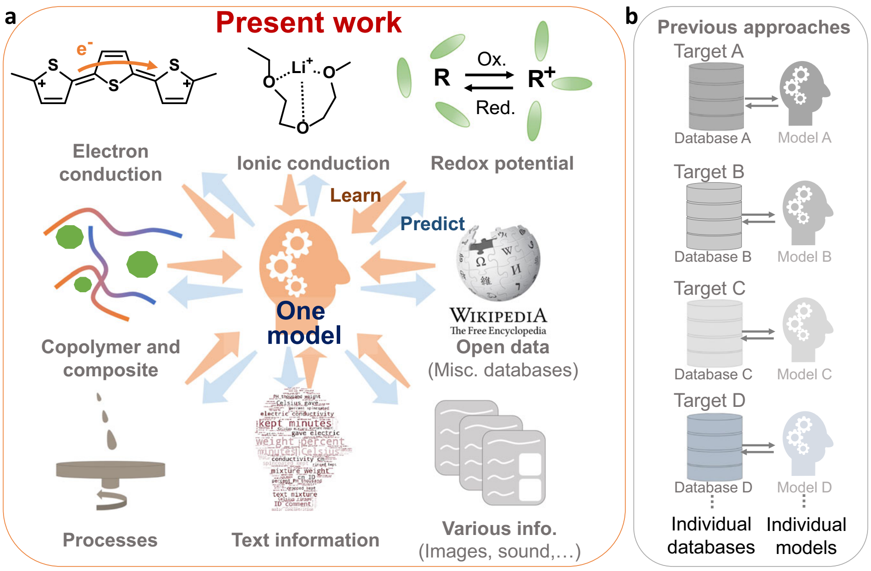
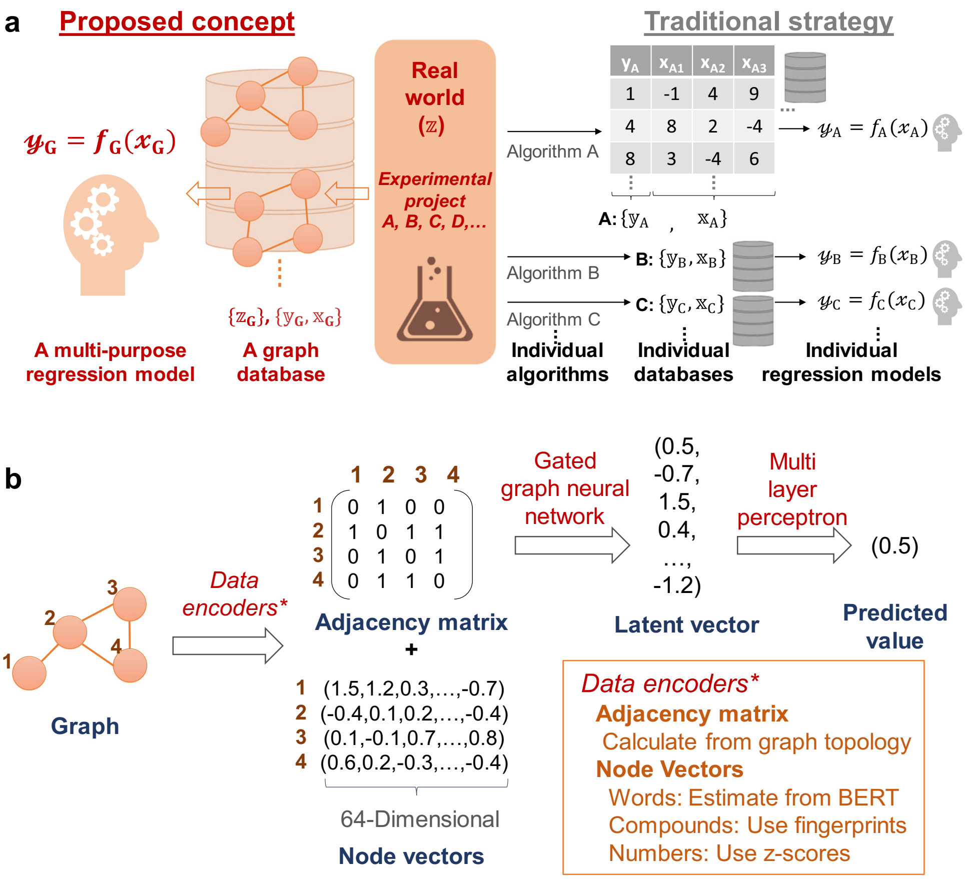


Fig. 1 Concept for the present work. a A single neural network model is trained to learn and predict diverse materials informatics projects. **b** Traditionally, a model can process only one database and predict one parameter. The Wikipedia logo is reprinted with the CC-BY-SA 3.0 license.



y_A	x_{A1}	x_{A2}	x_{A3}
1	-1	4	9
4	8	2	-4
8	3	-4	6

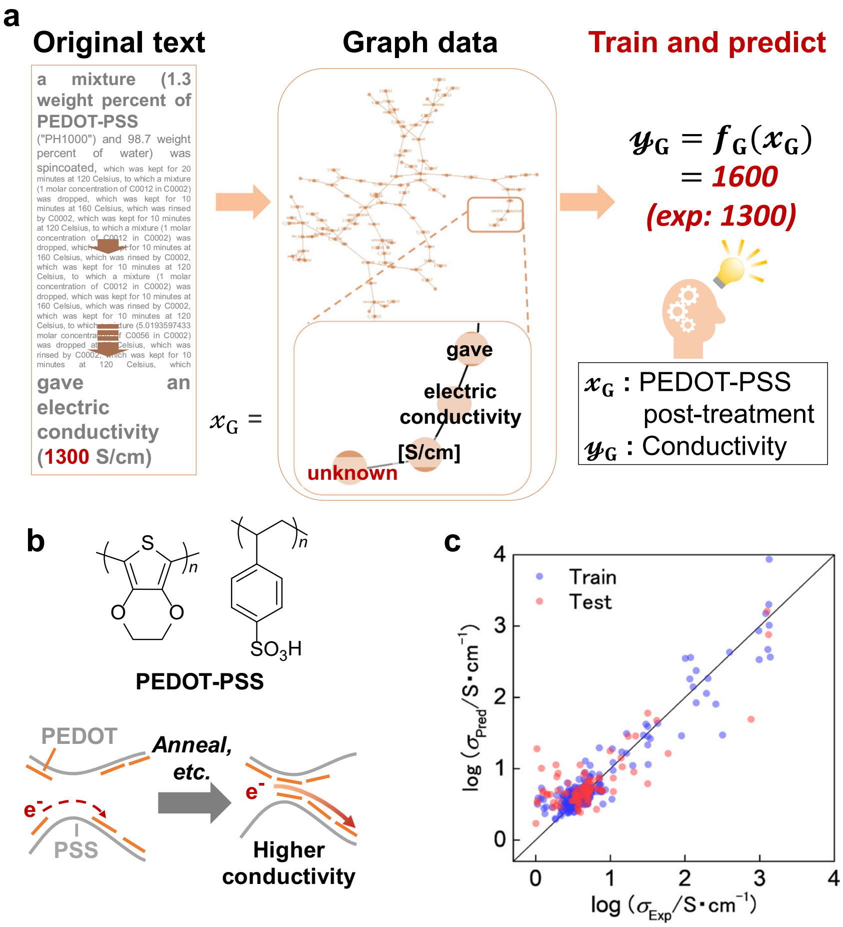
$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 0 & 1 \\ 4 & 0 & 1 & 1 & 0 \end{array}$$

1 (1.5,1.2,0.3,…,-0.7)
2 (-0.4,0.1,0.2,…,-0.4)
3 (0.1,-0.1,0.7,…,0.8)
4 (0.6,0.2,-0.3,…,-0.4)

(0.5, -0.7, 1.5, 0.4, …, -1.2)

*Data encoders**
Adjacency matrix
Calculate from graph topology
Node Vectors
Words: Estimate from BERT
Compounds: Use fingerprints
Numbers: Use z-scores

Fig. 2 About graph approach. a Concept of using a graph database for materials informatics. Traditionally, the experimental, real-world information (Z) is converted to table databases manually, which require unique algorithms to be formatted as numeric arrays for each case (right part of the figure). Machine learning models, trained with individual format databases, cannot interpret the other databases. On the other hand, common-formatted graphs were made to express versatile experimental information in this study (left part). A single machine learning model interprets all inputted experimental information. **b** Overview of processing graph structures for machine learning (see Supplementary Fig. 4 for further information).



a mixture (1.3 weight percent of PEDOT-PSS ("PH1000") and 98.7 weight percent of water) was spincoated, which was kept for 20 minutes at 120 Celsius, to which a mixture (1 molar concentration of C0012 in C0002) was dropped, which was kept for 10 minutes at 160 Celsius, which was rinsed by C0002, which was kept for 10 minutes at 120 Celsius, to which a mixture (1 molar concentration of C0012 in C0002) was dropped, which was kept for 10 minutes at 160 Celsius, which was rinsed by C0002, which was kept for 10 minutes at 120 Celsius, to which a mixture (1 molar concentration of C0012 in C0002) was dropped, which was kept for 10 minutes at 160 Celsius, which was rinsed by C0002, which was kept for 10 minutes at 120 Celsius, to which a mixture (5.0193597433 molar concentration of C0056 in C0002) was dropped at [illegible] Celsius, which was rinsed by C0002, which was kept for 10 minutes at 120 Celsius, which gave an electric conductivity (1300 S/cm)

Fig. 3 Prediction of electric conductivity. **a** Predicting the electrical conductivity of PEDOT-PSS from its post-treatment method. **b** Structure of PEDOT-PSS. The mixture gives higher conductivity with appropriate post-treatment, which induces the optimum connection of conductive moieties. **c** Experimental and predicted conductivity of PEDOT-PSS. Seventy percent of data are selected randomly as the training dataset. The rest is used only for prediction. R^2 scores for training and test datasets are 0.87 and 0.70, respectively.

reusing it in other projects may be difficult because it does not contain the context for the values (Supplementary Fig. 1).

In the present study, text-based experimental procedures were automatically converted to graphs while maintaining the text context and inputted directly to a machine learning model (Fig. 3a and Supplementary Fig. 3). Graph data describe the relationships among things (expressed as nodes) by connecting them with edges; in contrast to table databases, this flexibility enables the expression of diverse information easily, such as text structures, social networks, and molecular structures[28]. A recent development in deep learning has enabled the automatic recognition of the graphs and calculation of their characteristic features[28]. Still, individual prediction models had to be prepared for each genre. Here, we demonstrate that even a single graph and a prediction model can process multidisciplinary information, including chemicals, text, and numbers (Fig. 1).

A simple yet powerful approach to describe versatile information in graphs was to record both genre and content information in each node (Fig. 2b, Supplementary Figs. 4 and 5a). The original text of the post treatments was converted to dependency trees as undirected graphs by a natural language parser[29]. The nodes in graphs were classified as words, chemical structures, and values. The node information was then converted to numeric vectors by three algorithms (see "Methods" and Supplementary Fig. 4). To process words, a state-of-the-art language-understanding deep-learning model called BERT[30] was used. Molecular information was converted to vectors by using molecular fingerprints[31]. Values in numeric nodes were kept unchanged. To distinguish the genres, three numeric arrays were added to the headers. Apart from the three classes, any information can be embedded in graphs if it can be converted to numeric arrays, thereby paving the way to learning general information about materials science (e.g., inorganic structures, images, and sounds)[3].

More than 350 types of graphs related to the post-treatment of PEDOT-PSS (from over 20 papers) were prepared and inputted to a graph neural network[6,32]. The model was trained to predict the electrical conductivity from the post-treatment methods of the polymer films. In the original database, the procedures were written as text (Fig. 3a). After automatically converting the text to graphs, only the nodes containing electrical conductivity were replaced with the keyword "unknown". Here, no significant information is basically lost during the graph conversion because the quasi-reversibility of text parsing[29]. The graphs were used as the inputs (questions) to the model. The model was trained to predict the conductivity from the graph-shaped questions. As the overall user interface, the model can answer the performances of the polymers only from text. This style is more convenient and reliable for most users; special effort and knowledge are needed to prepare traditional table databases, which require the careful, manual selection of important features for the target phenomena and formatting into numeric arrays for machine learning.

The prediction accuracy of the conductivity by the neural network was high. To check the accuracy, the database was split into training (70%) and test (30%) datasets randomly. Although the model was trained only with the training dataset, the R^2 score of the prediction and experimental conductivity was greater than 0.7 (Fig. 3c). The score was comparable or slightly higher than the control experiment, where conductivity was predicted directly from texts using a conventional natural language model ($R^2 =$ 0.66, Supplementary Fig. 5b, c). The high accuracy supported the validity of the graph approach. Except for a few specialists, such accurate predictions are difficult to make due to the excessively complex preparation procedure (see the long preparation method shown in Fig. 3a). Because the neural network can find essential features from graphs automatically, manual parameter selection to prepare the database is not necessary. Automatic text parsing[29] and the general graph approach enable automatic data collection from materials science big data, where recognition of unstructured data has been a bottleneck[5].

Multitask learning in different projects. A key advantage of using general graphs is their high capability for describing diverse experimental information. Because the text context is maintained in the graphs, users can easily change the target parameters of the prediction by replacing the target node with the keyword "unknown". In contrast to normal table databases, the graph questions themselves contain the information about what is to be predicted. This enables one model to learn and predict multiple databases and parameters easily (multitask and multimodal training). For instance, we prepared a graph database containing more than 1000 chemical compounds from Wikipedia (Supplementary Fig. 6). The relationships among chemical structures and their physical properties were recorded as graphs. Similarly, a lithium-ion conducting polymer database, which we constructed previously[6], was converted into graphs. In the previous study, a long script was needed to process the complex conductor information so it could be interpreted by a machine learning model (i.e., into numeric arrays)[6]. However, in the present study, no additional script was necessary because the conductor information could be expressed in the general graph format (Supplementary Fig. 6).

Chemical graph databases were easily converted to question graphs by replacing the property nodes with the keyword "unknown". More than ten properties, including ionic conductivity, melting point, pKa, viscosity, and vapor pressure, were set as questions (Supplementary Fig. 6). A machine learning model was trained with the PEDOT-PSS and Wikipedia databases to predict the recorded properties (Fig. 4a and Supplementary Fig. 7). Multitask and multimodal training is not feasible with the traditional table databases, due to their inflexible format; the process information about PEDOT-PSS and the chemical properties in Wikipedia cannot be described fully in an integrated table.

Although there was no obvious relationship between the post-treatment of PEDOT-PSS and compounds in Wikipedia, the prediction accuracy of the electrical conductivity of PEDOT-PSS was improved by multitask training. The PEDOT-PSS database was split into training and test datasets randomly with different split ratios (0–0.9). All data from Wikipedia were combined with the training dataset (Fig. 4a). As expected, the R^2 score for the test dataset increased as the split ratio increased (Fig. 4b). Most importantly, the scores were always higher when Wikipedia was learned simultaneously. A similar improvement was observed for the multitask learning of the lithium-ion conducting polymer database. The score was more than three times higher with the multitask learning than with only learning PEDOT-PSS, with a split ratio of 0.3 (corresponding to learning ca. 100 cases of PEDOT-PSS). To our knowledge, this is the first report of multitask learning of different databases and improvement of prediction accuracy in materials science.

To reveal the detailed process of multitask learning, we analyzed the intermediate calculation steps in the model, by visualizing the outputs of a hidden layer in the neural network (Fig. 4c, Supplementary Figs. 8–11). The hidden layer converted the inputted graphs to 32-dimensional numeric arrays as the vector representation (Supplementary Fig. 5a). The vectors contain essential information about the input and output, termed 'latent space'[3,13]. We compressed the 32-dimensional vectors into two-dimensional arrays[33] for easier understanding (Fig. 4d). When only the Wikipedia database was used for training (split ratio of 0), the plots from PEDOT-PSS and Wikipedia databases were separate, indicating that the model interpreted them as different species. In contrast, the plots were combined after multitask learning because the model found hidden mutuality among the data and partially shared calculation algorithms for predicting the different parameters. Further mechanism analysis of the multitask is not accessible due to the "black box" problem of deep learning[3]. However, the recent idea of machine learning, represented by influence functions[34], may help researchers reveal the internal processes (e.g., clarify the relationships among specific databases and parameters).

A similar idea to multitask learning, called 'transfer learning', has also been proposed to improve prediction[6,8], in which different prediction models partially shared the calculation steps to recognize important data features efficiently. However, the final calculations were done by individual algorithms (for details see Supplementary Fig. 12). The individuality limits the advantages of the synergistic effects of learning multiple databases and acquiring broad knowledge of the field. In contrast, in the graph approach, a single intelligence interpreted multiple databases and properties. This finding is essential for exploring the materials informatics of experimental projects, most of which have limited database capacity owing to the high experimental cost[6].

General materials informatics prediction model. A more general materials informatics prediction model was pursued by increasing the number of learning databases. From public data, we collected 14 experimental materials science databases, containing over 40 properties (see Supplementary Information). The main compounds in the databases were monomeric molecules, organic polymers, and their mixtures. In addition to their basic physical properties, advanced features, such as redox potentials were included (Fig. 1). Prediction of redox potential is necessary to develop energy-related devices but was not fully successful, mainly because the potentials are changed by the effects of solvents and salts and complex systems are difficult to handle in the table format and simulations[35]. In the graph databases, the redox potentials were easily recorded as a function of redox molecules, solvents, and electrolyte salts.

For machine learning, some databases were selected randomly and learned with a single prediction model (Supplementary Fig. 13). By increasing the number of training databases, the number of predictable parameters increased because the model could understand the larger amount of information inputted. When the model was trained with all 14 databases, it could predict over 40 properties with high accuracy (Fig. 5, Supplementary Fig 14a, Supplementary Table 1, and Supplementary Data 1). The prediction accuracy was not high enough ($R^2 < 0$) with the parameters with insufficient amounts of training data (typically less than 100 cases, Supplementary Fig. 14b). We emphasize that the prediction errors by the multitask training were basically smaller than the control experiments, introducing

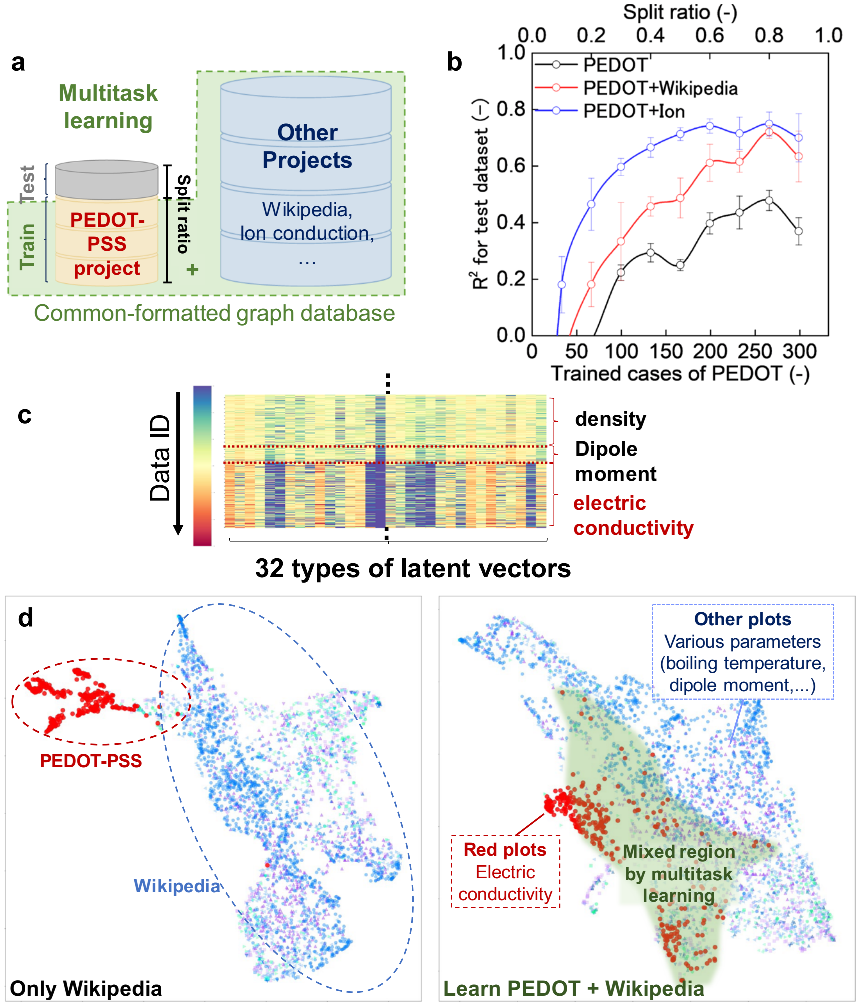
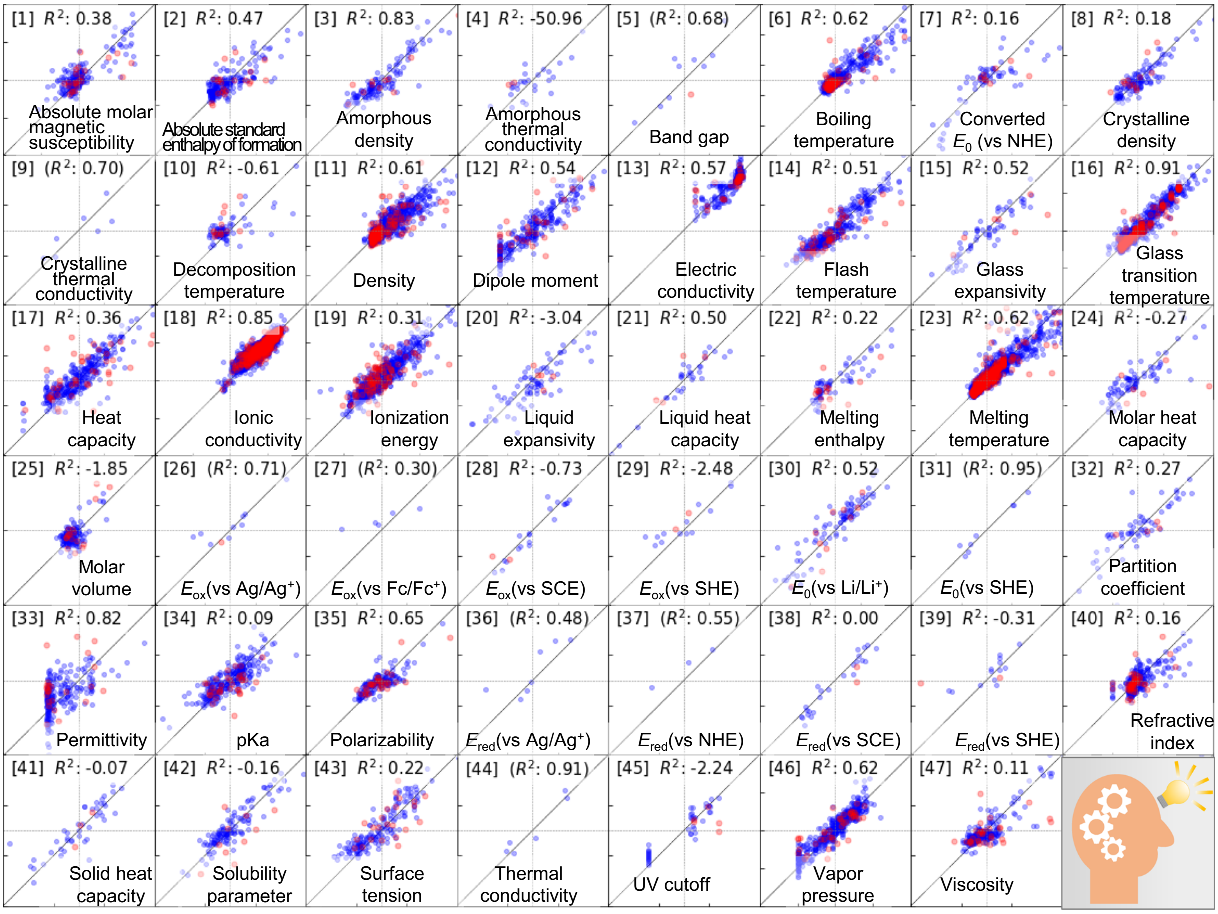


Fig. 4 Multitask learning. a Preparation of datasets for multitask learning. If the split ratio is 0.6, 60% of the randomly selected PEDOT-PSS data and all of the other database are used to train the model. The other 40% is selected as the test dataset. **b** R^2 scores for the test dataset after multitask learning with Wikipedia or the lithium-ion conducting polymer database. Only the PEDOT-PSS database is trained as a control. The values show the averages of five experiments with different random shuffling ($n = 5$ independent experiments). Error bars show the standard errors. See Supplementary Fig. 7 for the results of the training dataset. **c** Representative heatmap of the latent vectors for the PEDOT-PSS and Wikipedia databases, outputted by a model trained only with the PEDOT-PSS database. See Supplementary Fig. 9 for further information. **d** Two-dimensional UMAP projection of the latent vectors shown in Supplementary Figs. 9 and 10.

different random forest regressors to predict each parameter from corresponding chemical fingerprints (i.e., standard single-task training, Supplementary Fig. 14c). For higher accuracy, we are integrating other public experimental databases and even computational results. Together with revealing the synergetic effects of multitask learning, even one- or zero-shot learning[36] may be achievable with the model, which can benefit from both human-like context understanding capability and hugecomputational power to process big data. General prediction models will be beneficial to the wider research community because of their broad knowledge and ability to answer unknown questions; internet search engines can only answer questions about known issues and human professional resources are limited.

Inverse problem solving by graph approach. One of the ultimate goals of materials informatics is fully solving inverse problems. Instead of predicting the results of conditions carefully specified by humans, machine learning models are expected to answer much more ambitious questions, such as “Which post-treatment protocol for PEDOT-PSS will yield a conductivity of 10^4 S cm^{-1}?” or “What is the organic polymer structure that gives a melting point of 500 °C?”. Although there are no available procedures or structures that give high performances, integration of big scientific data may find the answers. In contrast, many informatics challenges must be overcome. The main difficulties are related to uniqueness of mapping, common sense, and generation of complex answers (Supplementary Fig. 15). For example, there would be multiple polymer structures exhibiting a melting point of the desired value. Therefore, an inverse function of melting point to structure cannot be determined uniquely (uniqueness problem). Furthermore, most candidate structures must be excluded automatically based on common sense from the field, namely by filtering out inappropriate compounds, for example, synthetically challenging or unstable compounds. Finally, generating complex information, represented by chemical structures, is still an open question in deep learning[13].

A general prediction model with the graph approach may be the key to solving inverse problems in materials science. Here, a graph neural network was trained with all the information from the 14 databases. In the previous section, nodes of numeric values (material properties) were set as the targets for prediction, whereas all types of nodes (number, word, and compound) in



Predict >40 properties using a single model

Fig. 5 Prediction of versatile parameters. Experimental (*x*-axis) and predicted (*y*-axis) values of each parameter after multitask learning of the 14 databases (90% of data was used for training). Blue and red plots correspond to the training and test datasets, respectively. The values are shown as *z*-scores. R^2 scores for test datasets are displayed. If test cases were not enough for evaluation, train scores are shown with brackets instead. Statistical results are also summarized in Supplementary Table 1. Larger graphs are shown in Supplementary Fig. 14a.

each graph were selected for prediction here (Supplementary Fig. 16). Final answers were constructed by finding the most similar vectors in the text and compound databases with the predicted values (see "Methods"). In the future, it may be possible to generate completely new answers by integrating an auto-encoder[13]. After training, the model could predict the original text in the graphs with a high accuracy of 96% (Supplementary Table 2). Even the 4% failed predictions were close to the answers (Supplementary Table 3). Typical mistakes were predicting "electrical conductivity" instead of "ionic conductivity" (answer) and "melting point" instead of "boiling point" (answer). This was because the model could understand the similar meanings of natural language by neural networks[30], resulting in near-miss answers.

The prediction accuracy of compounds was lower (36%), mainly because of the uniqueness problem (Supplementary Fig. 16). For instance, a chemical structure with a specific density (1.58 g cm^{-3}) and a melting point (203 °C) was questioned. The answer was "trehalose"; however, there should be other structures satisfying these two conditions. This problem with uniqueness lowered the apparent accuracy of compound prediction. In practical use, users can add optional conditions freely for desirable compounds. The structure will be determined uniquely if other properties such as heat capacity, hydrophilicity, and chemical stability are specified. Because machine learning models have no actual experimental experience, including this type of tacit knowledge that researchers have will be the key to ensuring the quality of predictions.

Discussion

We used a graph format to express diverse materials informatics information. This common format allows databases from different projects to be combined easily. A single neural network interpreted miscellaneous information, including chemical structures, more than 40 material properties, and text. Multitask and multimodal learning was essential not only for increasing prediction accuracy, but also for developing general-purpose prediction models for materials science. Integrating big data and improving the inverse problem-solving methods will allow this method to be used as an artificial materials science expert, which will change the traditional research and development cycle.

Methods

General information. Databases were constructed or collected from public data. All experimental data collected from the literature were converted into undirected graphs using original Python 3 scripts. Graph nodes were classified into three types: values, text, and chemical compounds, which were converted into numeric arrays automatically by different algorithms. All graph edges were treated equally. To train a neural network, values of target nodes (=*y*) in graphs were replaced with a specific keyword "__unknown__". The generated graphs were inputted as *x* to the model.

Databases. The PEDOT-PSS database was constructed in this work. The lithium-ion-conducting polymer database was constructed in our previous work[6]. The experimental properties of various compounds were collected from Wikipedia (https://ja.wikipedia.org/), Wikidata (https://www.wikidata.org/), Computational Chemistry Comparison and Benchmark DataBase (CCCBDB, https://cccbdb.nist.gov/), chemical suppliers, and the literature. All data were collected and rechecked by the authors.

Computer. Data processing and machine learning were conducted on a desktop computer (Intel Core i9-9900K CPU @ 3.60 GHz, 32 GB memory, GeForce RTX 2080 graphical processing unit, and Ubuntu 16.04 operating system).

Graph preparation from text. Text information about post-treatment of PEDOT-PSS was collected and converted to graphs by the following procedures (Fig. 3). We extracted the related experimental procedures from original articles (mainly experimental sections) and summarized them as text. The text was written in a set format for ease of machine learning (e.g., avoiding inconsistent spelling). In the future, automatic text collection (probably by machine learning) will be examined, to reduce the cost and to eliminate the human nature of data preparation. On the other hand, we note that the system can be robust against orthographical variances and different text expressions owing to the language-understanding deep-learning model[30] (as long as sufficient data are given).

In the text, compounds were expressed as their IDs, such as C0001 and C0123, and their structure information was recorded in a compound database. In the database, the compound ID and simplified molecular input line entry system (SMILES) expressions were associated (about 50 types of chemicals). Numeric values were standardized as *z*-scores by each unit (e.g., siemens per centimeter and degrees Celsius, Supplementary Fig. 3). The text was parsed automatically by an open module (StanfordNLP 0.2.0)[29] to construct dependency trees of words. Nodes of less important words and symbols (e.g., “at”, “were”, “and”, “was”, “by”, and “to”) were removed after parsing. Nodes of conductivity values (*y*) in graphs were replaced with “__unknown__” to prepare the inputs (*x*) for machine learning. An open-source library (NetworkX 2.4, https://networkx.github.io/) was used to generate graphs.

Graph databases for multitask learning. For multitask learning, all databases, typically written as tables, were converted to graphs by Python scripts if necessary. In the graphs, the relationships among the factors were connected by edges (Supplementary Fig. 1d). As a common rule, a numeric parameter was connected in the order “parameter name”—“value”—“[unit]” to describe a property of a target node. Numeric values were standardized as the z-score by each unit (e.g., degrees Celsius and siemens per centimeter). For multitask learning, a graph database of PEDOT-PSS was reconstructed manually (apart from the text database described in the previous section) because the automatic parser made graphs according to a different rule (Supplementary Fig. 3). A free graph editor (yEd 3.19) was used to draw graphs in the “graphml” format, which was compatible with the NetworkX library.

An integrated compound database was made by combining the compound information in each database and a chemical supplier’s catalog (Tokyo Chemical Industry Co.). A total of over 29,000 chemical structures were recorded. The integrated database was used for the multitask learning experiments (i.e., except for the automatic text parsing experiment in Fig. 3 and Supplementary Fig. 8). The prediction score can decrease slightly when a larger compound database is used (e.g., compare R^2 scores in Figs. 3c and 4b) because of the larger loss of compound information after converting into 64-dimensional numeric vectors with larger data. Therefore, the compressing algorithms should be improved in future research.

Converting graphs to numeric matrices and vectors. For machine learning, graphs were converted into adjacency matrices and numeric vectors (Supplementary Fig. 4a). Adjacency matrices, expressing the node connections, were simply calculated by a function of NetworkX. A matrix, *D*, was prepared according to the following steps: (1) assign unique IDs to each node; and (2) determine D_{ij}, which is 1 if nodes *i* and *j* were connected, and otherwise is 0.

Each node content was converted to 64-dimensional numeric arrays (Supplementary Fig. 4b). Other information, such as images and sounds, can be also treated by implementing additional processing scripts (see below). The first four-dimensional arrays of the vectors were headers to distinguish the node types. Three types of different random numeric arrays were assigned using an embedding function of Chainer 7.2.0, an open library for deep learning. The remaining 60-dimensional arrays were prepared by three different algorithms according to node types.

Numeric nodes: When the node represents a number, the corresponding 60-dimensional numeric array will be a repeat of the value. For instance, an array of (0.5, 0.5,..., 0.5) is set for the value node of 0.5.

Text nodes: To process text nodes, a natural language recognition model (BERT[30], pretrained model of “uncased_L-24_H-1024_A-16”, accessible at https://github.com/tensorflow/models/tree/master/official/nlp/bert) was employed. The model could calculate 768-dimensional numeric vectors of the corresponding words, phrases, and text. Similar text expressions or meanings were converted to similar vectors by BERT[30]. The 768-dimensional vectors were compressed to 60-dimensional numeric arrays using a principal component analysis (PCA) algorithm, implemented with an open-source library (scikit-learn 0.22.2).

Compound nodes: Organic compounds were recorded as unique IDs (e.g., C0023) in graph nodes. By referencing a compound database, their structure information, expressed by 60-dimensional numeric arrays, was loaded. In the compound database, molecular structures were recorded as SMILES expressed by character strings. Their chemical features were calculated using an open module of chemistry (RDKit 2019.03.2, https://www.rdkit.org/), to obtain extended-connectivity fingerprints of the 2048-bit data. The binary data were split every 4 bits (e.g., (0011), (0000), (1111), …) and converted to corresponding integers (e.g., 512-dimensional array of (3, 0, 15, …)). Finally, a 60-dimensional array was prepared followed by PCA compression and standardization.

Other nodes: In this study, only numeric, text, and molecular structure nodes were implemented in the graphs. In the future, other information, such as inorganic crystal structures, images, sounds, and spectra, will be processed by adding appropriate scripts to express them as vectors.

Dataset preparation. Target values (*y*) on the nodes in graphs were replaced with a keyword “__unknown__” to prepare problems (*x*) automatically (underlines are added to distinguish from the word “unknown”). If one graph has more than two target values (e.g., melting point and density), the replacement and problem generation were done individually to generate multiple problems; there were not multiple “__unknown__” nodes in one graph. The numeric nodes of the material properties were set as the target values (*y*) for prediction. For inverse problem solving in the last section of the main manuscript, all nodes (numbers, chemicals, and text) in all graphs were set as problems.

Unless noted otherwise, the graph data were split into train and test datasets randomly (splitting ratio of 0 to 0.9). The train dataset was used to train a graph neural network and the test dataset was used only for prediction.

Machine learning. The prepared datasets were trained with a neural network (Supplementary Fig. 5). The Chainer library was used to script the model, which had a graph neural network layer to recognize graphs and three dense layers to calculate the final outputs. The graph neural network was prepared based on an open-source library in Chainer-chemistry 0.7.0. The Implemented function of the gated graph neural network was used. Only the input part of the function was modified to input the adjacency matrices and node vectors described above, whereas the original version was customized to input only the connection of atoms in molecules.

The neural network was trained to reduce the mean square errors between the predicted and actual values. Minibatch sizes of 32 and 128 were selected for only PEDOT-PSS learning and multitask learning, respectively. Training was repeated with 100 epochs with the Adam optimizer[37]. The dimension of output values was 1 for the normal prediction mode of numeric nodes.

The model was constructed with a 64-dimensional output to solve inverse problems in the last section of the main manuscript. The last 60-dimensional vectors were used for prediction (i.e., the first four-dimensional arrays were used only to distinguish node types). The predicted vectors were compared with the word (or compound) list in the integrated databases. Ones giving the highest cosine similarity with the predicted vectors were extracted as the prediction result. In the future, direct outputs of words and compounds may be achieved using autoencoders or similar techniques[13]. Numeric nodes were predicted by averaging the predicted 60-dimensional vectors. All hyperparameters were optimized manually. Automatic parameter tuning will be tested in future research with higher computing power (e.g., multiple GPUs).

Prediction by conventional models as the control experiments. *Language model to predict conductivity directly from text* (*related to* Fig. 3 and Supplementary Fig. 5): In Fig. 3, conductivity was predicted via graph structures, which were converted from texts. On the other hand, conventional recurrent neural networks, such as long short-term memory (LSTM)[38], can treat text information directly. As a control experiment, conductivity was predicted from the texts. The conductivity values in the texts were replaced with “__unknown__” to make problems (Supplementary Fig. 5b). After converting words into embedding vectors, the word inputted to a LSTM layer (which outputs 16-dimensional latent vectors, implemented by Keras 2.3.1). Conductivity was calculated via a dense layer without activation functions.

Random forest regressors to predict chemical properties (*related to* Fig. 5 *and* Supplementary Fig. 14): As the control to the multitask training, machine learning was conducted in a conventional way with a Wikipedia database. Random forest was selected as a conventional yet robust prediction algorithm[4]. First, compound information was converted to 60-dimensional arrays through the same process as the graph approach. Then, individual random forest regressors (by scikit-learn) were introduced and trained to predict each chemical property recorded in the database (absolute standard enthalpy of formation, boiling temperature, decomposition temperature, density, flash temperature, ionization energy, melting enthalpy, melting temperature, refractive index, vapor pressure, and pKa) from the

60-dimensional arrays. Train and test datasets were prepared randomly with a splitting ratio of 9/1.

Data availability

Databases used for the analyses are available at https://github.com/KanHatakeyama/Integrating-multiple-materials-science-projects/ (https://doi.org/10.5281/zenodo.3910817). All related data that support the findings of this study are available from the corresponding authors upon reasonable request. Original data for Supplementary Table 1 are provided as an additional supplementary file (Supplementary Data 1).

Code availability

Source codes (i.e., prediction of conductivity from texts, multitask learning, inverse problems, and control experiments) and databases used for the analyses are available at https://github.com/KanHatakeyama/Integrating-multiple-materials-science-projects/ (https://doi.org/10.5281/zenodo.3910817).

Received: 18 April 2020; Accepted: 2 July 2020;
Published online: 30 July 2020

References

1. Bell, G., Hey, T. & Szalay, A. Computer science. Beyond the data deluge. *Science* **323**, 1297–1298 (2009).
2. Leonelli, S. Data—from objects to assets. *Nature* **574**, 317–320 (2019).
3. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
4. Ramprasad, R., Batra, R., Pilania, G., Mannodi-Kanakkithodi, A. & Kim, C. Machine learning in materials informatics: recent applications and prospects. *Npj Comput. Mater.* **3**, 54 (2017).
5. Hill, J. et al. Materials science with large-scale data and informatics: unlocking new opportunities. *MRS Bull.* **41**, 399–409 (2016).
6. Hatakeyama-Sato, K., Tezuka, T., Umeki, M. & Oyaizu, K. AI-assisted exploration of superionic glass-type Li(+) conductors with aromatic structures. *J. Am. Chem. Soc.* **142**, 3301–3305 (2020).
7. Kim, C., Chandrasekaran, A., Huan, T. D., Das, D. & Ramprasad, R. Polymer genome: a data-powered polymer informatics platform for property predictions. *J. Phys. Chem. C* **122**, 17575–17585 (2018).
8. Yamada, H. et al. Predicting materials properties with little data using shotgun transfer learning. *ACS Cent. Sci* **5**, 1717–1730 (2019).
9. Nakada, G., Igarashi, Y., Imai, H. & Oaki, Y. Materials-informatics-assisted high-yield synthesis of 2D nanomaterials through exfoliation. *Adv. Theory Simul.* **2**, 1800180 (2019).
10. Ito, K., Obuchi, Y., Chikayama, E., Date, Y. & Kikuchi, J. Exploratory machine-learned theoretical chemical shifts can closely predict metabolic mixture signals. *Chem. Sci.* **9**, 8213–8220 (2018).
11. Gomez-Bombarelli, R. et al. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nat. Mater.* **15**, 1120–1127 (2016).
12. Granda, J. M., Donina, L., Dragone, V., Long, D. L. & Cronin, L. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature* **559**, 377–381 (2018).
13. Gomez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
14. Jensen, Z. et al. A machine learning approach to zeolite synthesis enabled by automatic literature data extraction. *ACS Cent. Sci.* **5**, 892–899 (2019).
15. Hiszpanski, A. M. et al. Nanomaterial synthesis insights from machine learning of scientific articles by extracting, structuring, and visualizing knowledge. *J. Chem. Inf. Model.* https://doi.org/10.1021/acs.jcim.0c00199 (2020).
16. Aykol, M. et al. Network analysis of synthesizable materials discovery. *Nat. Commun.* **10**, 2018 (2019).
17. Mrdjenovich, D. et al. Propnet: a knowledge graph for materials science. *Matter* **2**, 464–480 (2020).
18. Tshitoyan, V. et al. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95–98 (2019).
19. Pei, J. et al. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* **572**, 106–111 (2019).
20. Ramachandram, D. & Taylor, G. W. Deep multimodal learning: a survey on recent advances and trends. *IEEE Sign. Process. Mag.* **34**, 96–108 (2017).
21. Kim, G. H., Shao, L., Zhang, K. & Pipe, K. P. Engineered doping of organic semiconductors for enhanced thermoelectric efficiency. *Nat. Mater.* **12**, 719–723 (2013).
22. Xia, Y., Sun, K. & Ouyang, J. Solution-processed metallic conducting polymer films as transparent electrode of optoelectronic devices. *Adv. Mater.* **24**, 2436–2440 (2012).
23. Bießmann, L. et al. Highly conducting, transparent PEDOT:PSS polymer electrodes from post-treatment with weak and strong acids. *Adv. Electron. Mater.* **5**, https://doi.org/10.1002/aelm.201800654 (2019).
24. Roch, L. M. et al. From absorption spectra to charge transfer in nanoaggregates of oligomers with machine learning. *ACS Nano* https://doi.org/10.1021/acsnano.0c00384 (2020).
25. Muckley, E. S., Collins, L., Srijanto, B. R. & Ivanov, I. N. Machine learning-enabled correlation and modeling of multimodal response of thin film to environment on macro and nanoscale using “Lab-on-a-Crystal”. *Adv. Funct. Mater.* **30**, https://doi.org/10.1002/adfm.201908010 (2020).
26. Tanaka, F., Sato, H., Yoshii, N. & Matsui, H. Materials informatics for process and material co-optimization. *IEEE Trans. Semicond. Manuf.* **32**, 444–449 (2019).
27. Takahashi, K. & Tanaka, Y. Materials informatics: a journey towards material design and synthesis. *Dalton Trans.* **45**, 10497–10499 (2016).
28. Zhou, J. et al. Graph neural networks: a review of methods and applications. https://arxiv.org/abs/1812.08434 (2018).
29. Qi, P., Dozat, T., Zhang, Y. & Manning, C. D. Universal Dependency Parsing from Scratch. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 160–170 (Publisher is Association for Computational Linguistics, Brussels, Belgium, 2018). https://doi.org/10.18653/v1/K18-2001.
30. Devlin, J., Chang, M.-W., Lee, K. & Toutanov, K. BERT: pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/abs/1810.04805 (2019).
31. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).
32. Li, Y., Tarlow, D., Brockschmidt, M. & Zemel, R. Gated graph sequence neural networks. https://arxiv.org/abs/1511.05493 (2017).
33. McInnes, L., Healy, J. & Melville, J. UMAP: uniform manifold approximation and projection for dimension reduction. https://arxiv.org/abs/1802.03426 (2018).
34. Koh, P. W. & Liang, P. Understanding Black-box Predictions via Influence Functions. https://arxiv.org/abs/1703.04730 (2017).
35. Jinich, A., Sanchez-Lengeling, B., Ren, H., Harman, R. & Aspuru-Guzik, A. A mixed quantum chemistry/machine learning approach for the fast and accurate prediction of biochemical redox potentials and its large-scale application to 315000 redox reactions. *ACS Cent. Sci* **5**, 1199–1210 (2019).
36. Socher, R. et al. Zero-shot learning through cross-modal transfer. https://arxiv.org/abs/1301.3666 (2013).
37. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. https://arxiv.org/abs/1412.6980 (2014).
38. Sak, H., Senior, A. & Beaufays, F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. https://arxiv.org/abs/1402.1128 (2014).

Acknowledgements

This work was partially supported by Grants-in-Aid for Scientific Research (Nos. 17H03072, 18K19120, 18H05515, 20H05298, and 19K15638) from MEXT, Japan. The work was also partially supported by a research grant from the Center for Data Science, Waseda University, and Information Services International-Dentsu, Ltd. The work was also partially supported by the Research Institute for Science and Engineering, Waseda University.

Author contributions

K.O. and K.H. conceived the project. K.H. conducted experiments and wrote the manuscript. All authors analyzed the data and contributed to the discussion.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s43246-020-00052-8.

Correspondence and requests for materials should be addressed to K.O.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2020