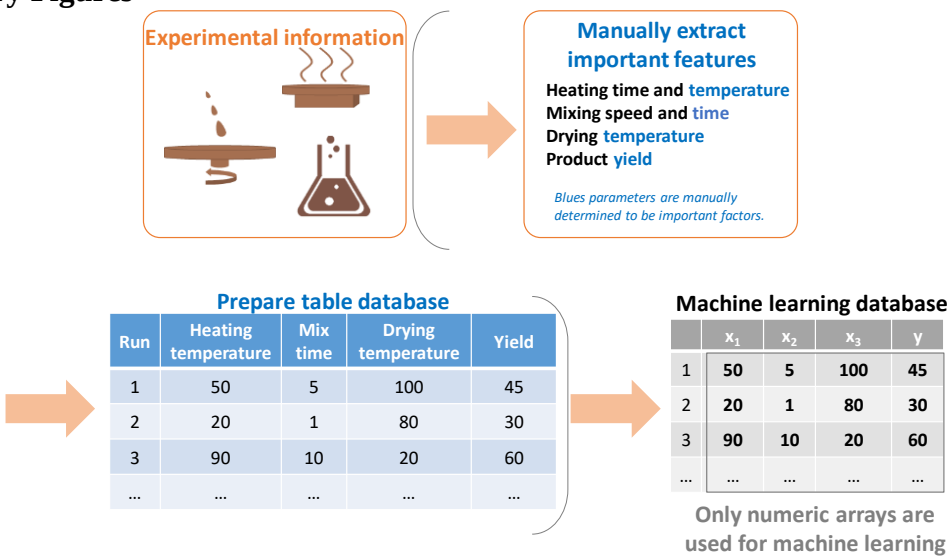
Supplementary Information

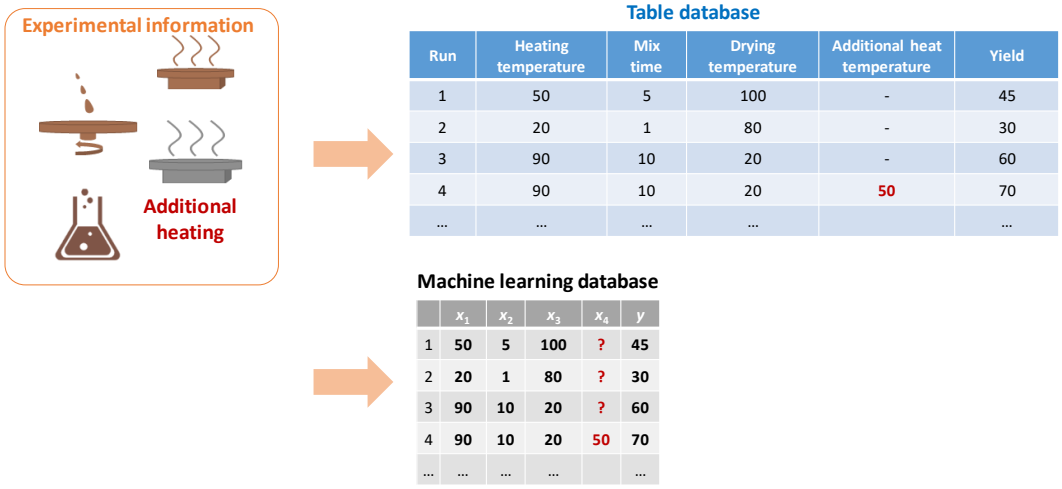
Integrating multiple materials science projects in a single neural network

*Kan Hatakeyama-Sato and Kenichi Oyaizu**

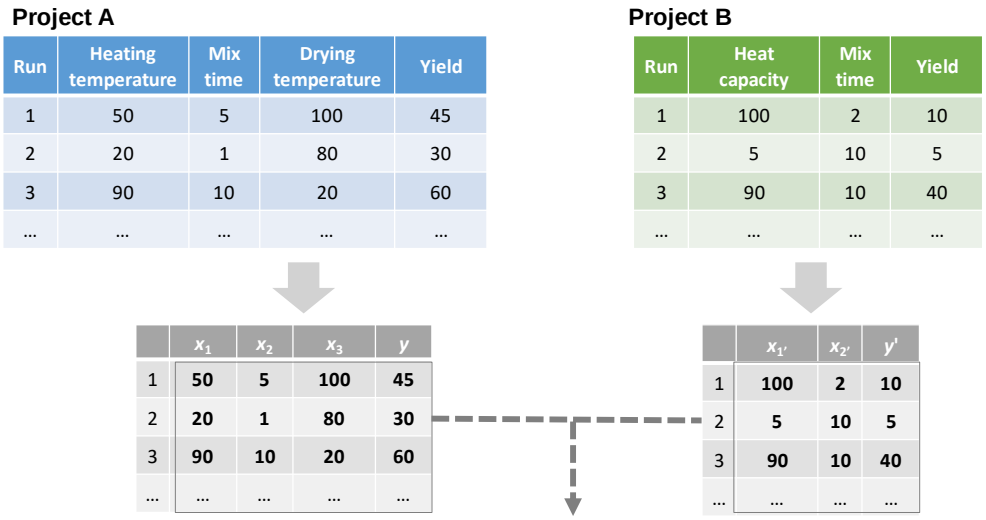
Supplementary Figures



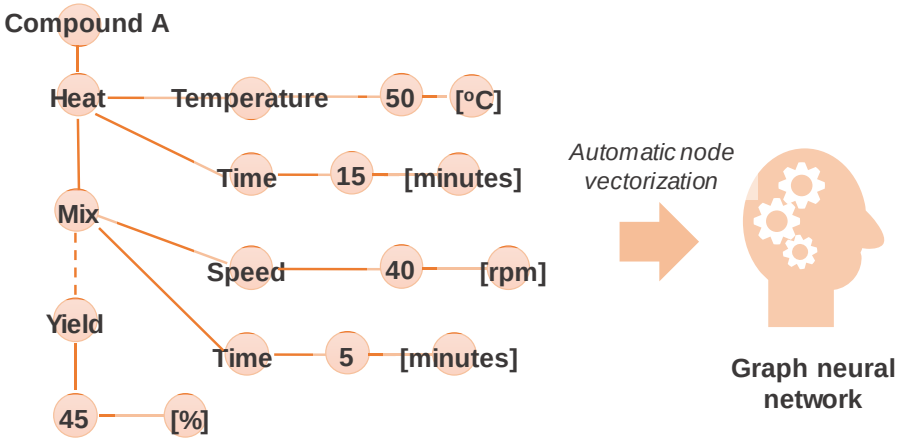
a



b



c



d

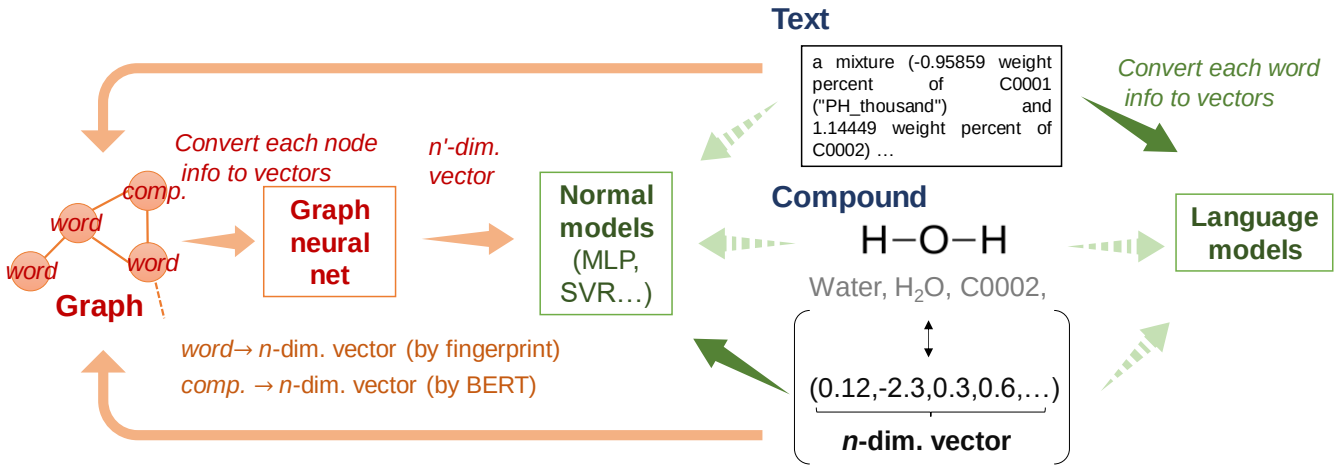
Supplementary Fig. 1. Comparison of table and graph databases.

a Typical algorithm for converting experimental information to numeric arrays. After selecting the supposedly important factors to affect the target value ("yield"), experimental data are summarized as a table. For machine learning, only numbers in the table are extracted.

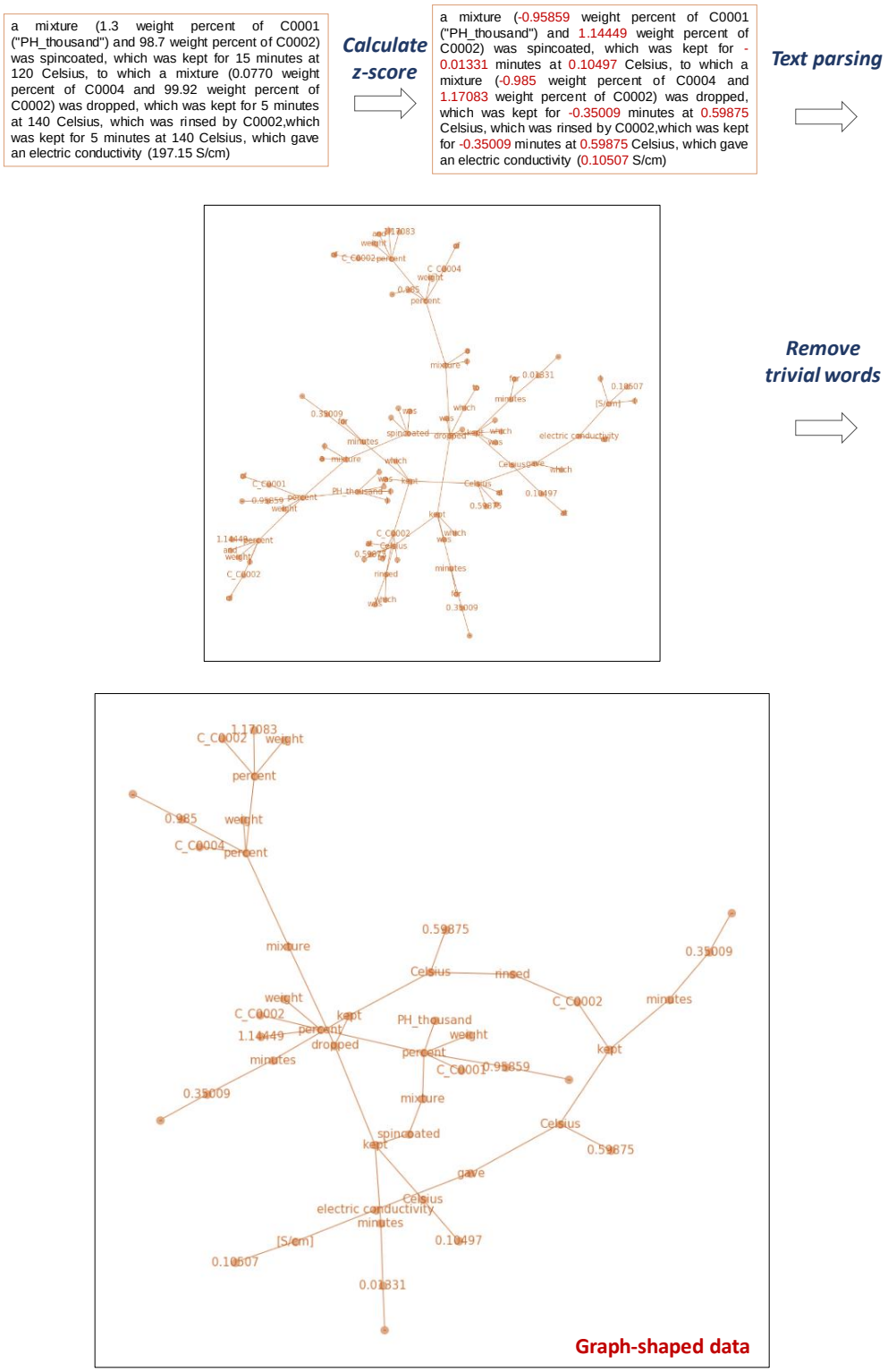
b Typical problem for table databases. If an additional heating step is found to be important for the target parameter, a column must be added to the database. In addition, the table becomes sparser after every addition of a new factor.

c Difficultly in integrating different table databases. The databases need to be reformatted because normal machine learning models accept only numeric arrays with a fixed format. However, there is no obvious way of combining tables in different formats. Furthermore, numeric arrays without text context have little information about the experiment. Therefore, the integration itself may not be beneficial to the models.

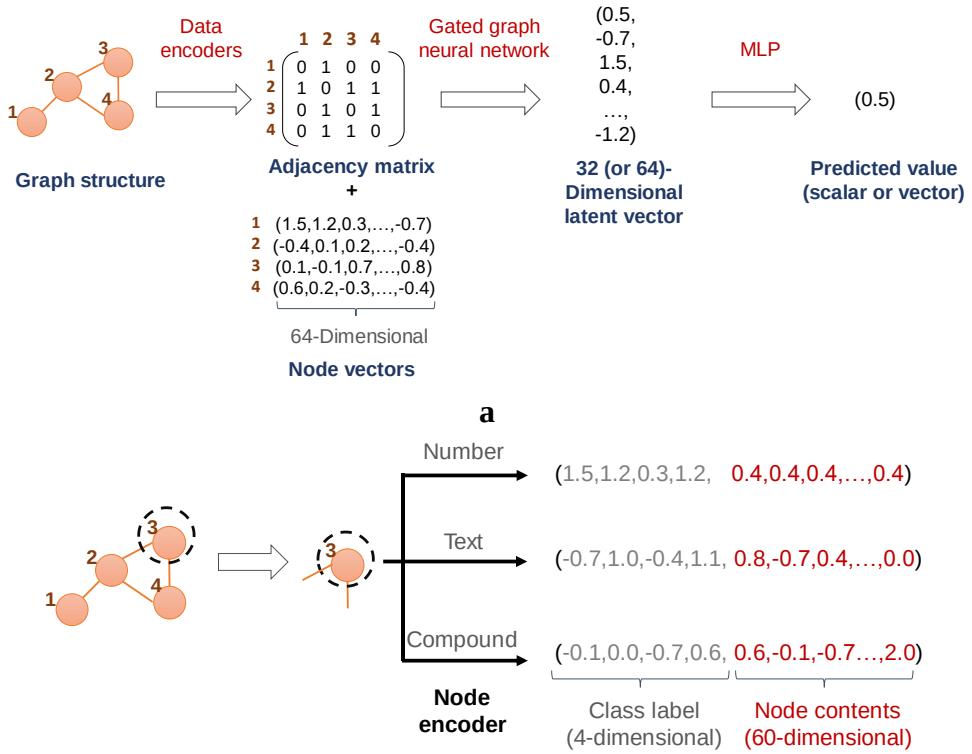
d Concept of the graph format for expressing experimental information. Like a flowchart, experimental procedures are described easily. By connecting new nodes, the characteristics of a target node can be explained in detail. In contrast to the table approach, the text context of numeric values is maintained in the format, which enables the effective multitask learning of different databases.



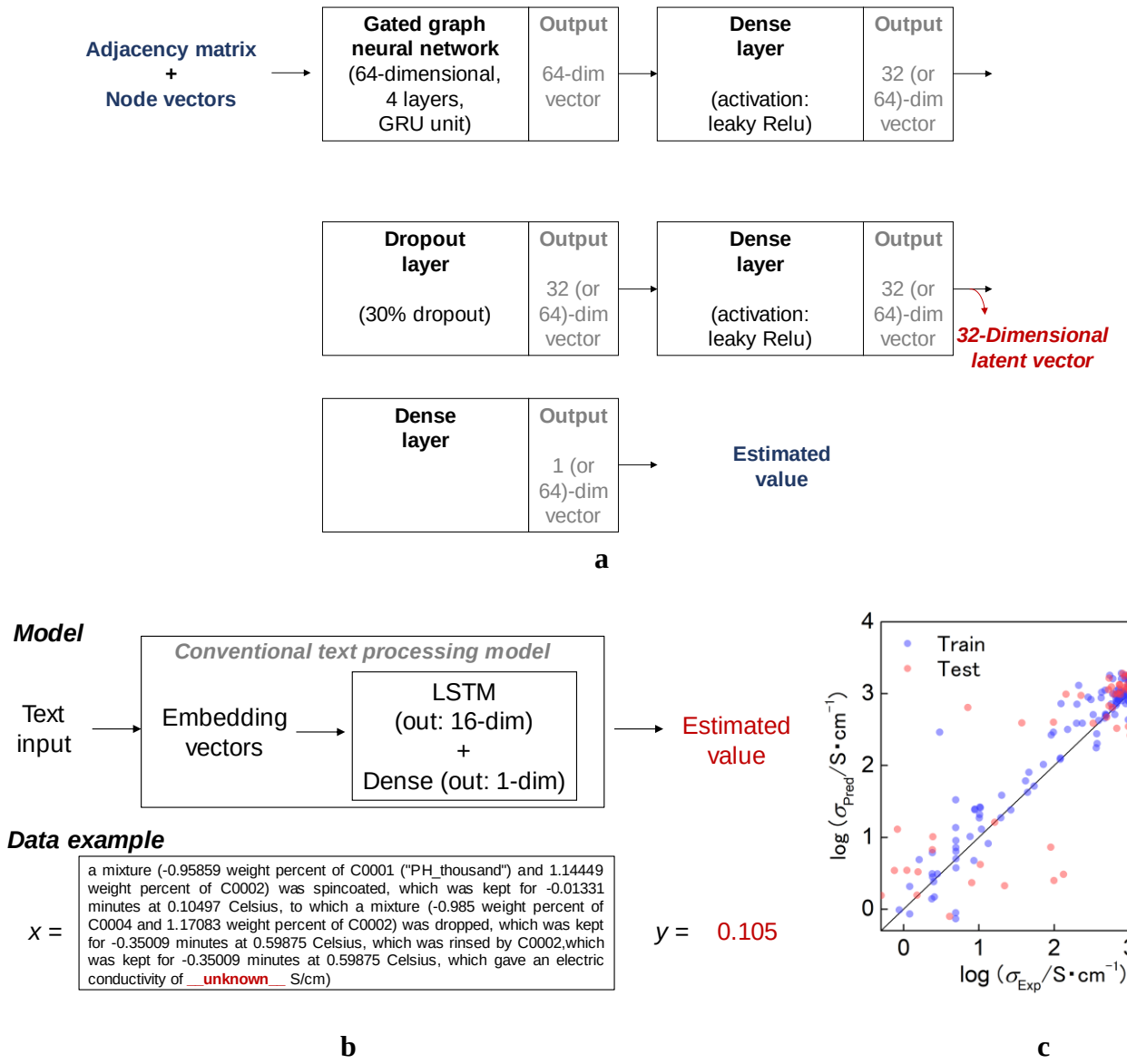
Supplementary Fig. 2. Comparison of our approach with the conventional ones (MLP: multi layer perceptron and SVR: support vector regressor).



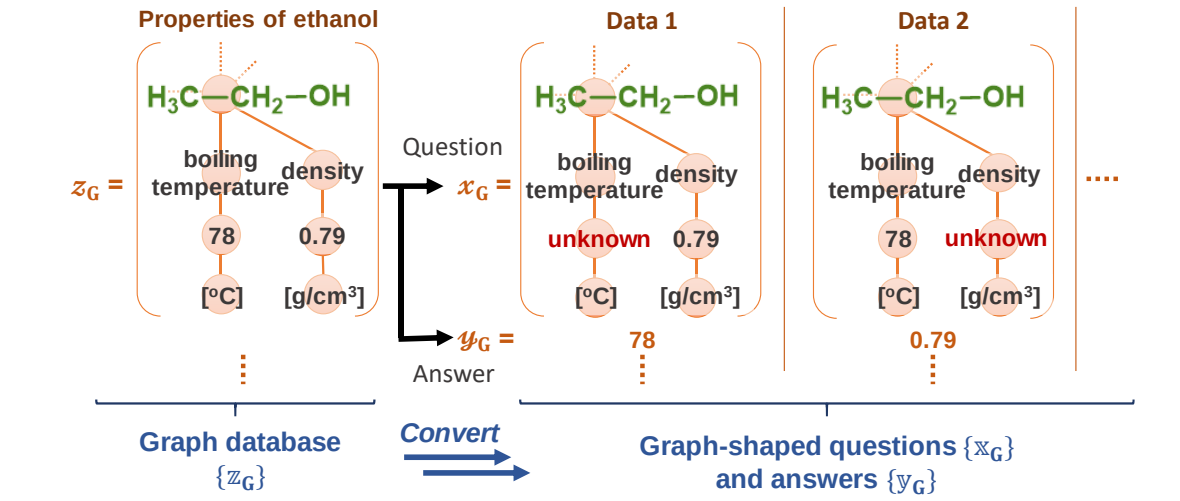
Supplementary Fig. 3. Scheme for preparing graph data from text automatically. Numbers are converted to z-scores, and then a dependency tree is generated as a graph by a natural language parser. Trivial nodes, such as “a”, are removed automatically.



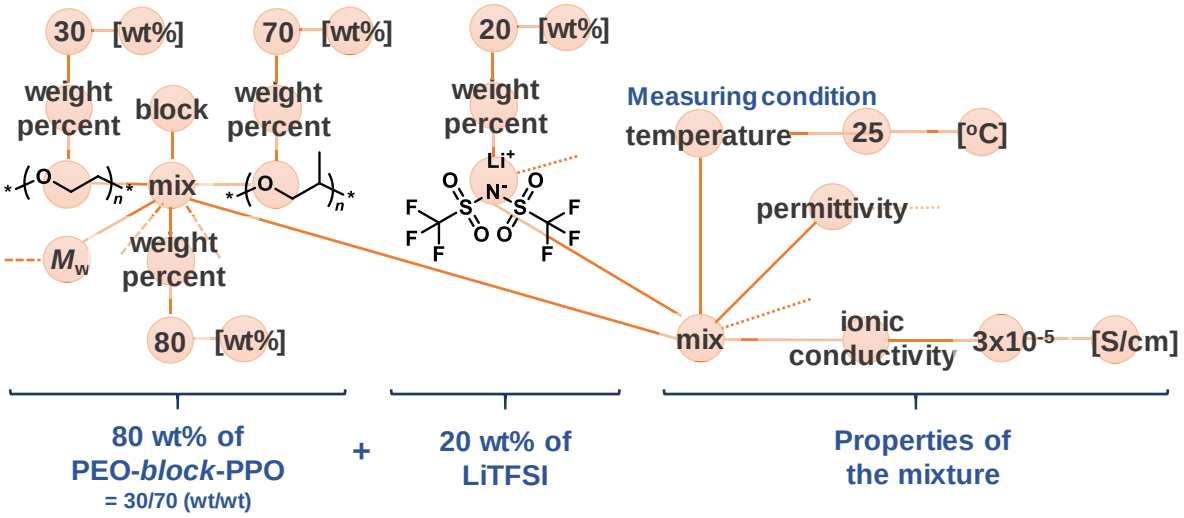
Supplementary Fig. 4. Overview of processing graph structures for machine learning.
a General scheme. The connections of nodes in a graph are expressed as an adjacency matrix. Information from each node is converted to 64-dimensional numeric arrays (node vectors). The matrix and vectors are inputted to a gated graph natural network and a multilayer perceptron (MLP). Finally, a scalar (or 64-dimensional vector) is calculated.
b Node encoder. Node information is converted to a 64-dimensional numeric array. The first four-dimensional array represents the node types. The remaining array represents the content.
c Algorithms to treat text and compounds. The information is converted according to BERT and fingerprint techniques, respectively.



Supplementary Fig. 5. Prediction of electric conductivity.
a Configuration of a neural network for interpreting graph information. The dimension of hidden layers is 32 for the normal mode of predicting scalars. The model has around 57,000 trainable parameters in total. The intermediate 32-dimensional output is extracted to visualize latent vectors in Supplementary Figs. 8–10. For the inverse problem solving, the dimension of output and hidden layers is set as 64.
b Configuration of a LSTM model to predict electric conductivity of PEDOT-PSS directly from the text, using LSTM.
c Prediction results by the LSTM model (control experiment to Fig. 3). Seventy percent of the data was selected randomly as the training dataset. The rest was used only for prediction. R^2 scores for training and test datasets were 0.94 and 0.66, respectively. The linguistic approach was as powerful as the graph with this dataset, but has inherent problems, especially with new chemicals.



a

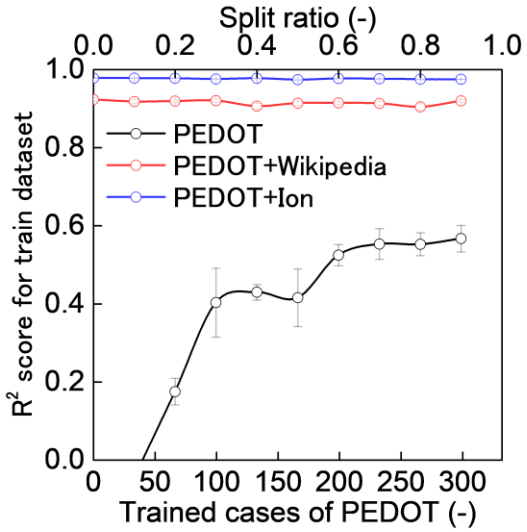


b

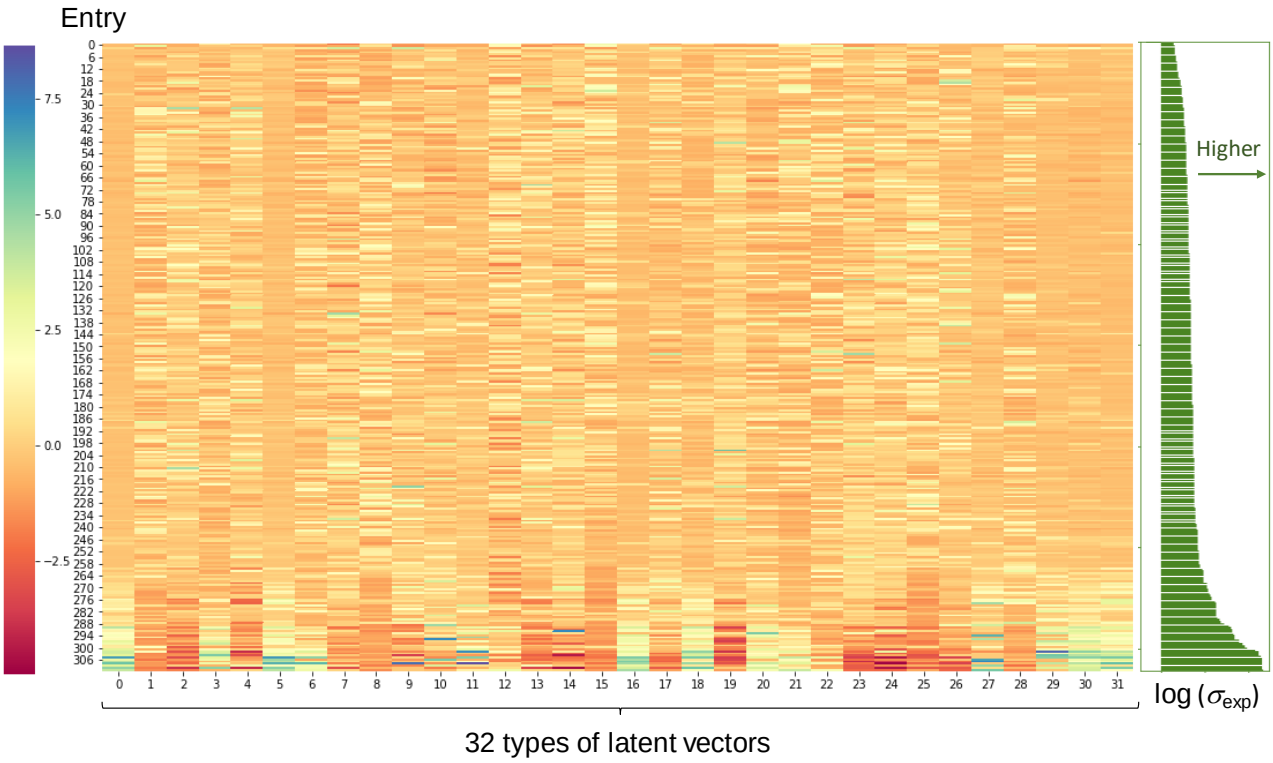
Supplementary Fig. 6. Examples of generating graph data questions from experimental information.

a Example of ethanol. In the database, ethanol has a boiling point of 78°C and density of 0.79 g cm^{-3}. The chemical structure node is connected with the property nodes (original graph z_G). To prepare a problem graph (x_G) for machine learning, one of the nodes is replaced with the keyword "__unknown__". The deleted value is set as the target for prediction (y_G)

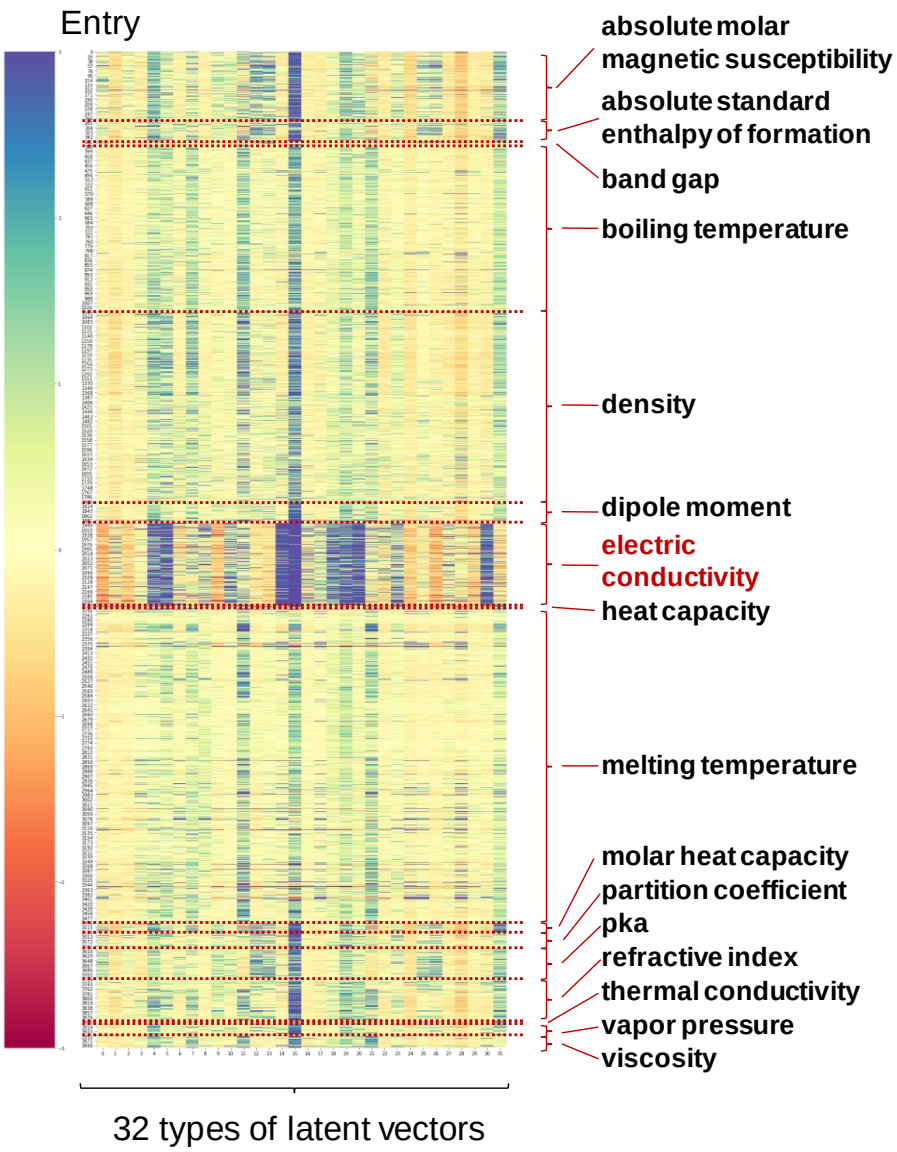
b Example of an ionic conductor as a mixture of chemicals. For a block polymer, the chemical structure nodes are connected with the node "mix". The additional keyword "block" is added to claim the block structure. The copolymerization ratio is recorded from the "weight percent" nodes. Additional polymer information, such as molecular weights is connected as the supplementary nodes. The physical mixing process is explained in a similar way to copolymers. The mutual word "mix" is used to express both copolymers and physical mixing.



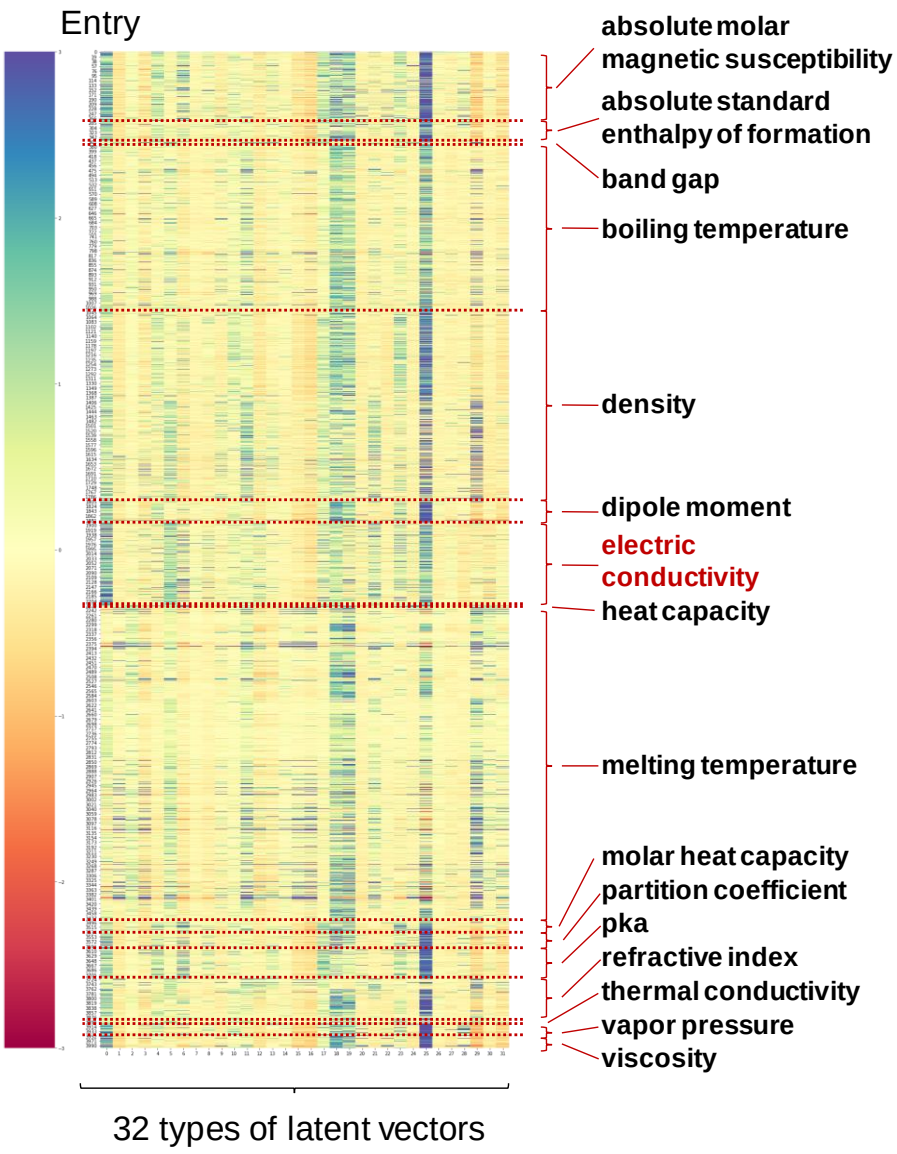
Supplementary Fig. 7. R^2 scores for the multitask learning of PEDOT-PSS and Wikipedia (or lithium-ion conducting polymer database). The model is trained with the random part of the PEDOT-PSS database with a different split ratio (0 to 0.9, Fig. 4a) and the whole of another multitasking database (Wikipedia or lithium-ion conducting polymer database). The R^2 scores for the predicted and experimental values with the trained datasets are shown (test dataset results are shown in Fig. 4b). The training scores are high with the multitask training because all cases of Wikipedia or the lithium-ion conducting polymer database were trained. The presented values are the averages of five random splittings of the database (n = 5 independent experiments). Error bars show standard errors.



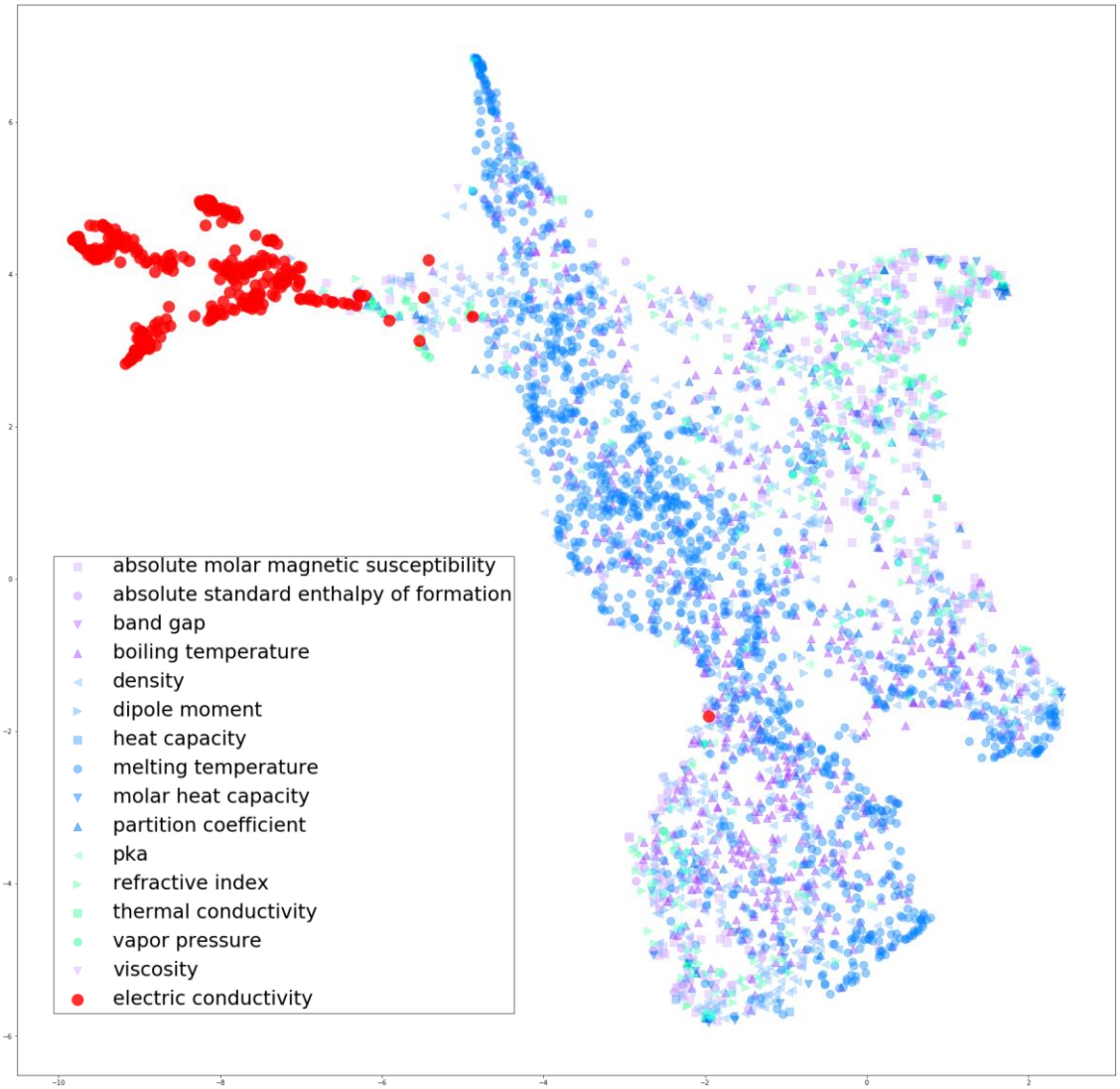
Supplementary Fig. 8. Heatmap of the latent vectors for the PEDOT-PSS database. After training a neural network with the PEDOT-PSS database (90% of the data was used for training), the 32-dimensional latent vectors (see Supplementary Fig. 5) for each piece of data are calculated. Data are sorted by the observed conductivity. The vectors become more characteristic with higher conductivity. This indicates that the prediction model recognizes important features of the high conductivity from the graphs.



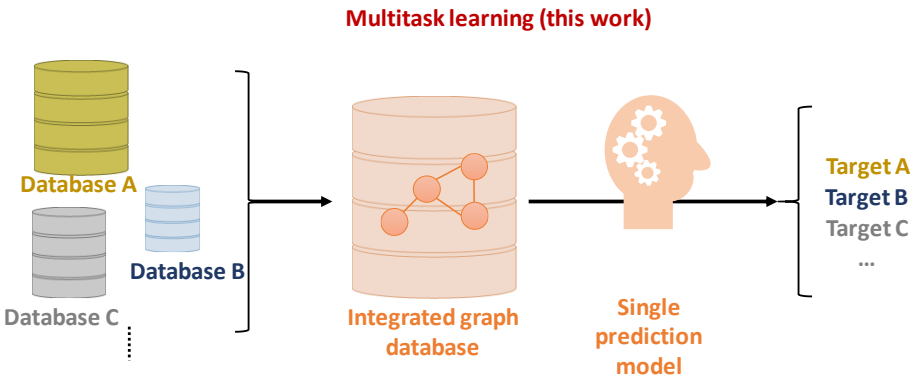
Supplementary Fig. 9. Heatmap of the latent vectors for the PEDOT-PSS and Wikipedia databases. The model is trained with only Wikipedia (i.e., split ratio of 0 in Fig. 4a). The latent vectors for the two databases are calculated in the same way as in Supplementary Fig. 8. The vectors change substantially only when the electrical conductivity of PEDOT-PSS is inputted. The difference is visualized more clearly by the two-dimensional mapping in Fig. 4a. The colour difference indicates that the electrical conductivity prediction is done in a different way from the other parameters.



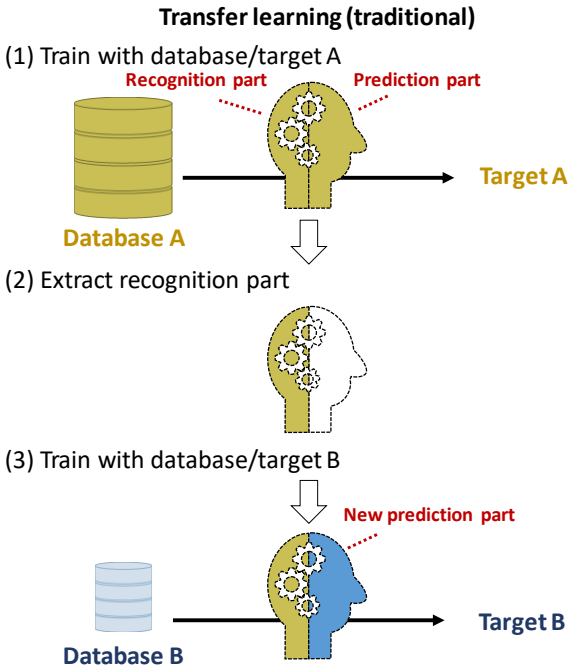
Supplementary Fig. 10. Heatmap of the latent vectors for the PEDOT-PSS and Wikipedia databases, after multitask learning of the both (split ratio of 0.9 in Fig. 4a). The vectors for electrical conductivity prediction are similar to the other properties. Separating those vectors is infeasible in Fig. 4d, indicating that the model does not have particular weights or algorithms to predict electrical conductivity.



Supplementary Fig. 11. Two-dimensional projection of Supplementary Fig. 10. This figure is a larger version of Fig. 4d in the main manuscript. Compression was done by an UMAP algorithm.[1]

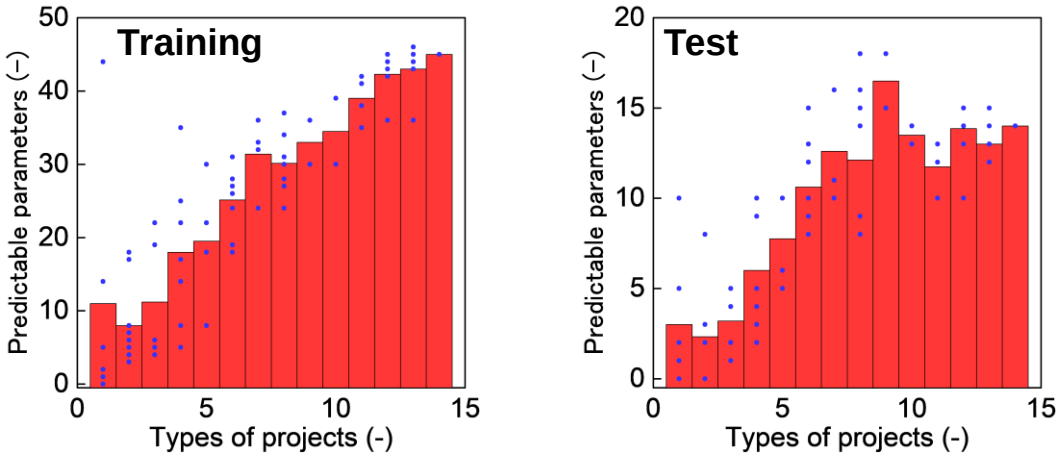


a



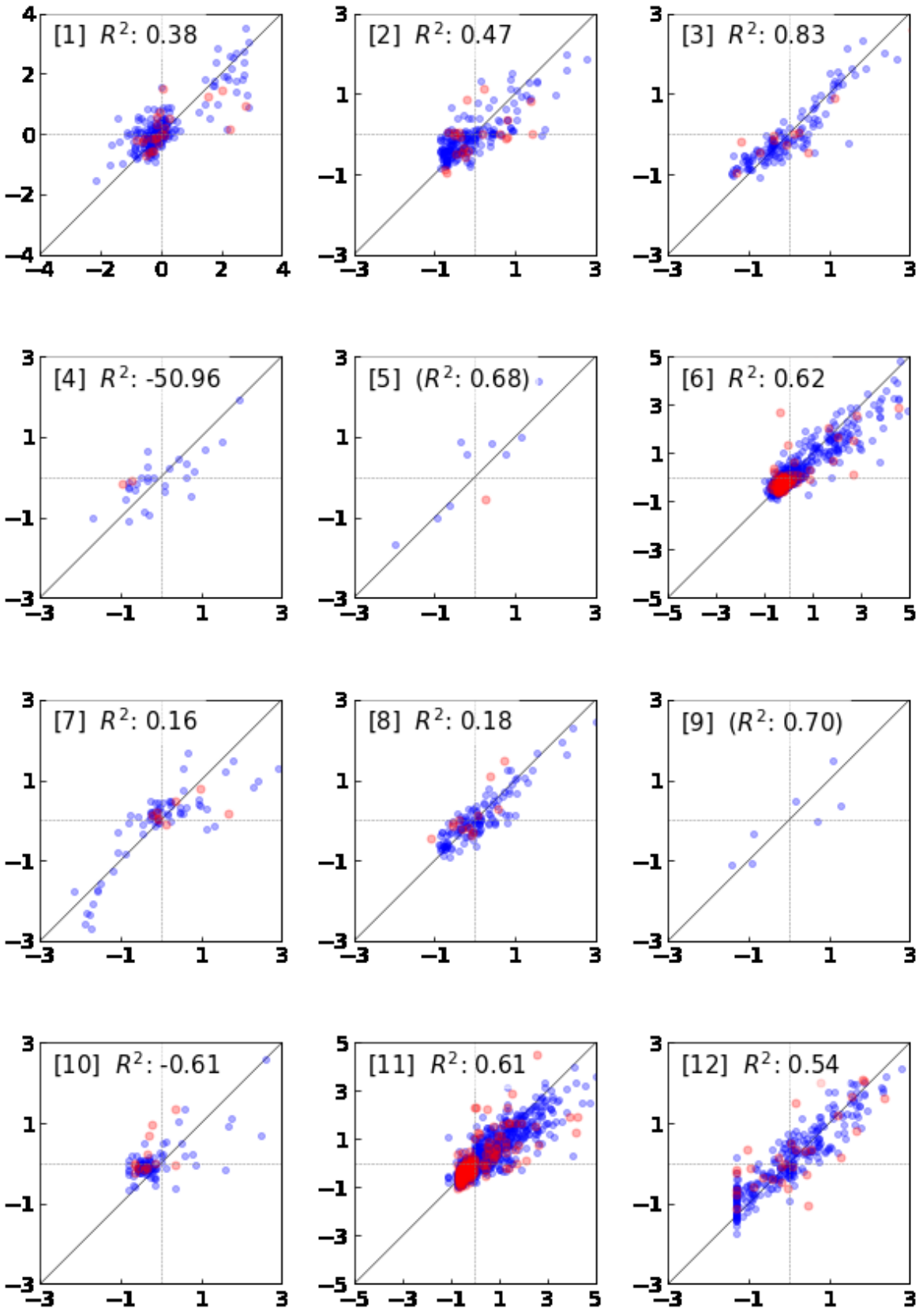
b

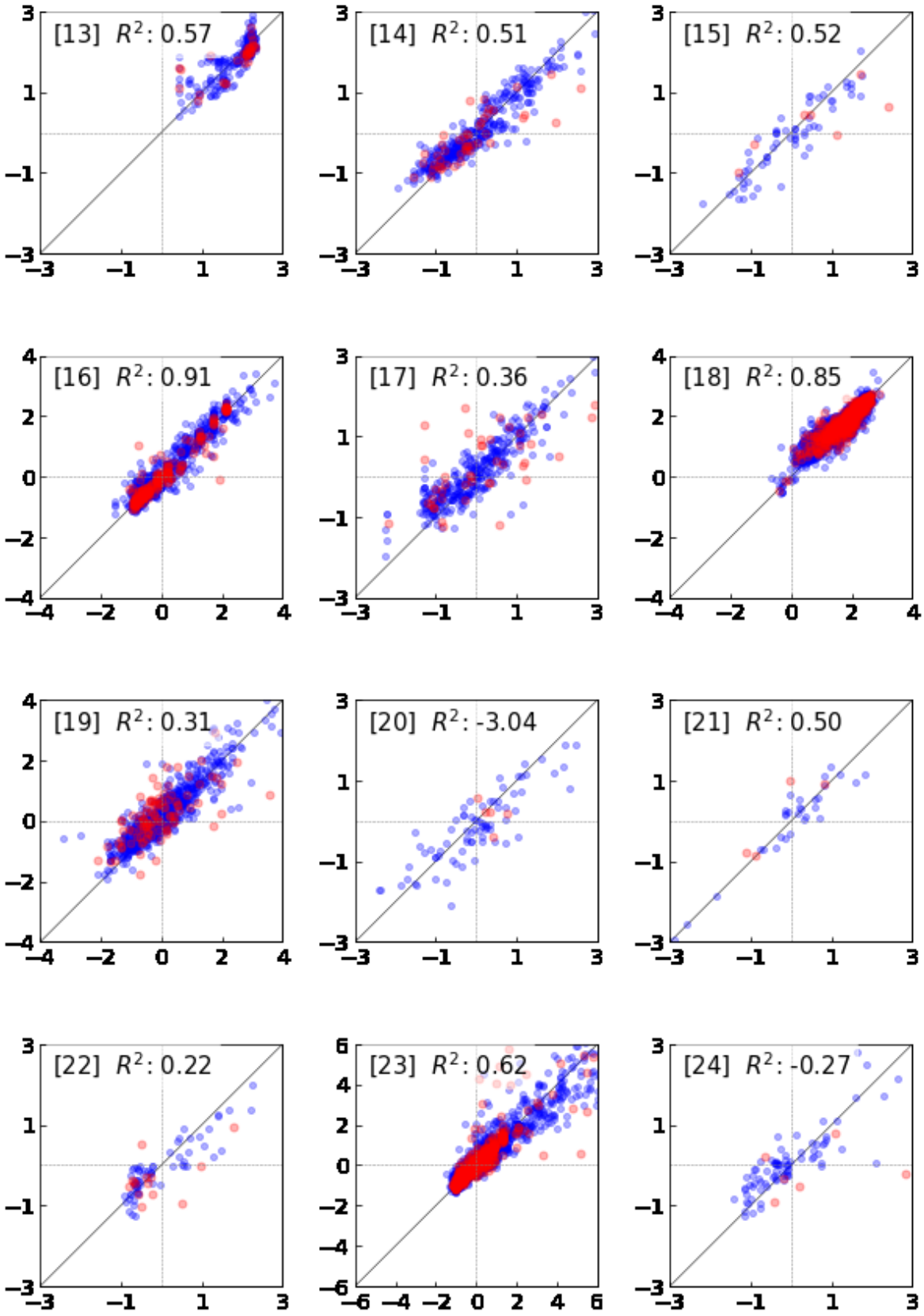
Supplementary Fig. 12. Comparison of multitask and transfer learning.
a Multitask learning. In the present work, all data are simultaneously trained with a single prediction model.
b Transfer learning. First, a prediction model is trained with a single (and typically large) database. The model has two main parts: the recognition of the characteristic features of data and the final calculation of the target values, performed by a grated graph neural network and the following dense layers in Supplementary Fig. 5, respectively. For transfer learning, only the recognition part is extracted and reused in a new model. It is trained with another (typically smaller) database. Normally, only the prediction part is trained. The pretrained part can be useful for recognizing the important features efficiently even in the new database. However, the final prediction part has no relationship with the former model. Therefore, in contrast to multitask learning, synergetic improvement of prediction accuracy does not occur.

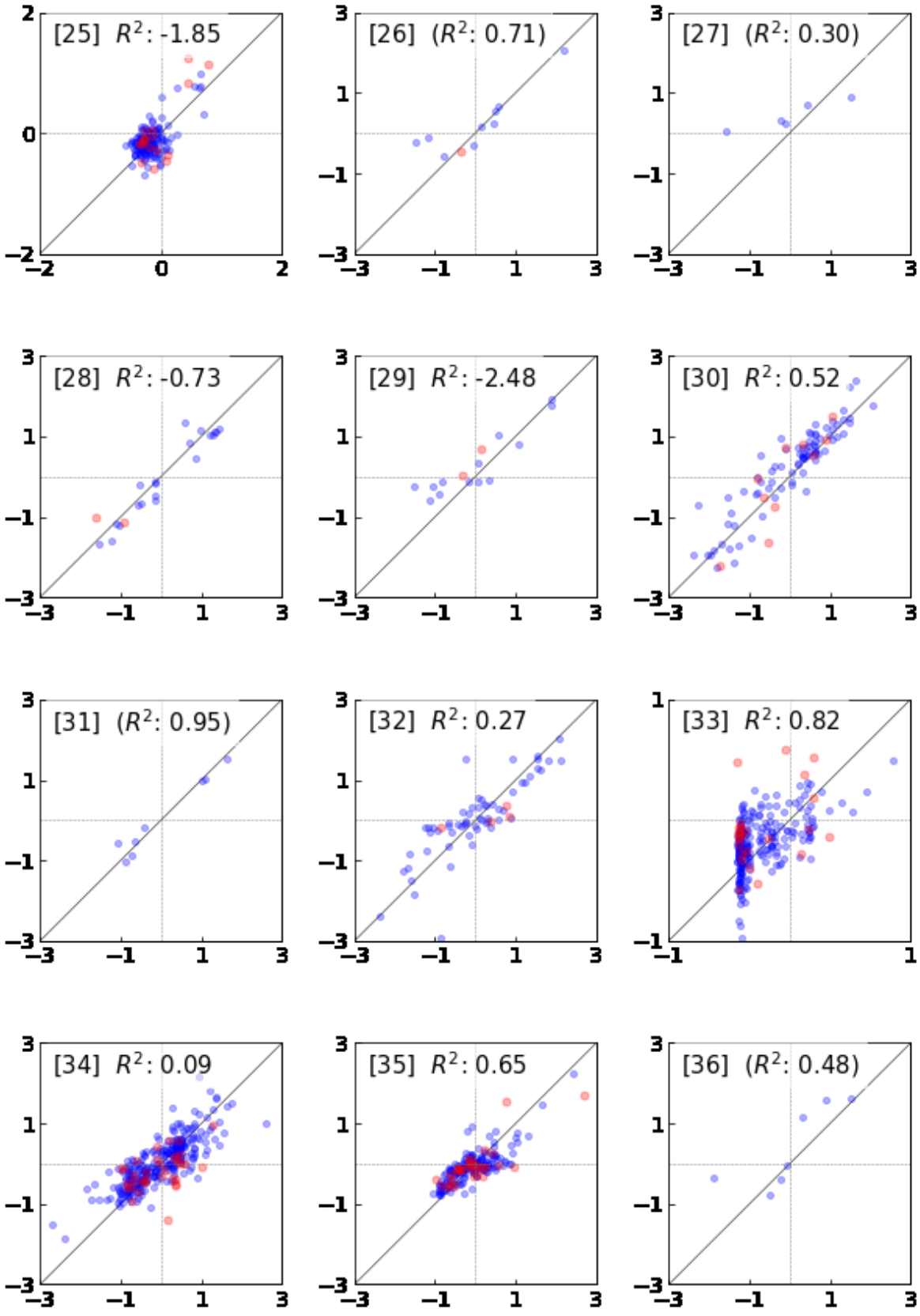


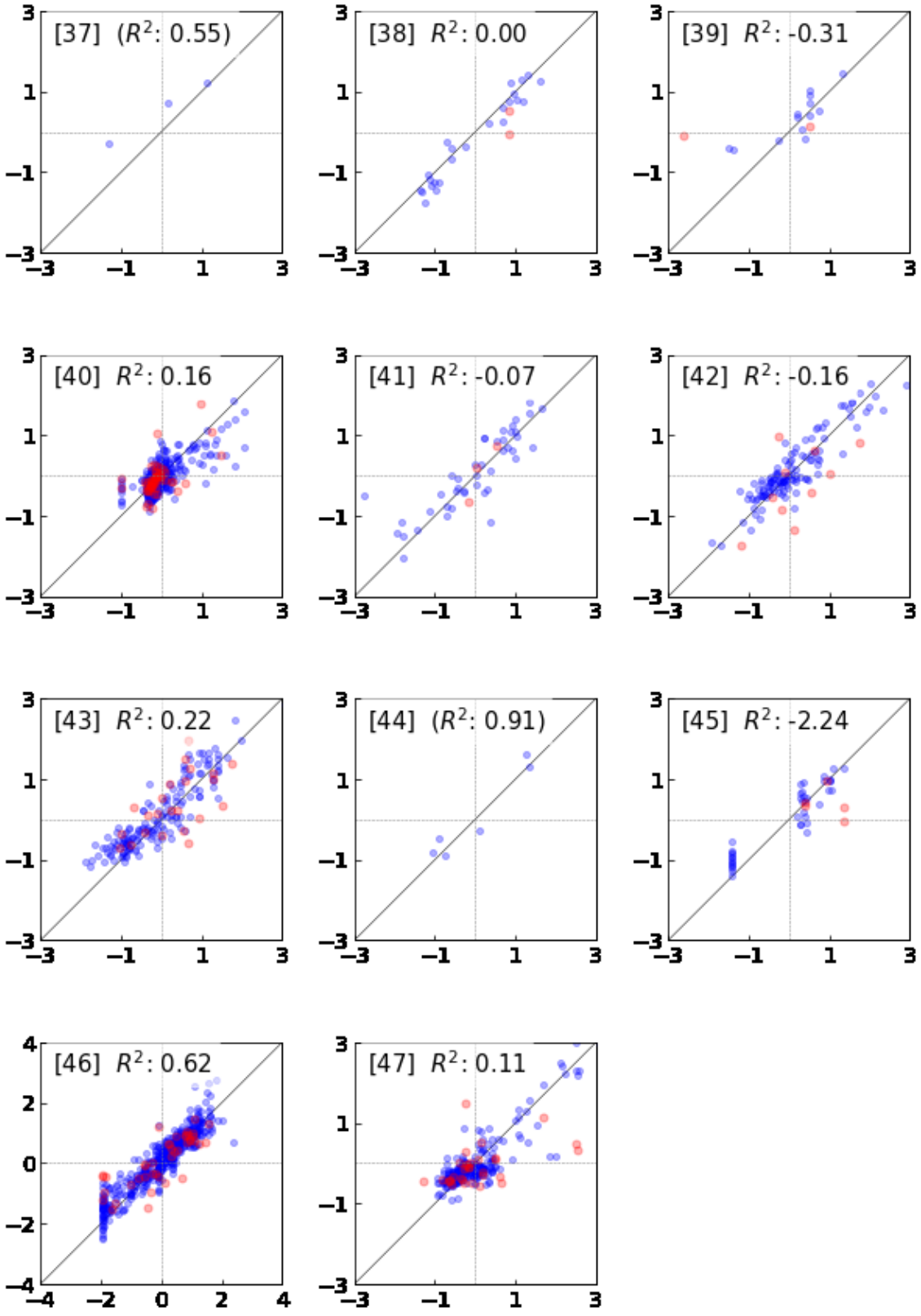
a b

Supplementary Fig. 13. Number of predictable parameters for the a training and b test datasets after training with randomly selected databases. First, several databases are selected randomly from the 14 types of databases. After splitting into training and test datasets (ratio of 9:1), the prediction model is trained. For each parameter, R^2 scores for the prediction and experimental values are calculated. Here, "predictable parameters" are defined as parameters with $R^2 > 0.3$. After repeating the random selection and training, bar graphs are plotted to show the relationships between the number of trained projects and average predictable parameters. Blue plots show the original values for each trial. The variance of the parameters mainly comes from the differences in the number of parameters and records in the databases.

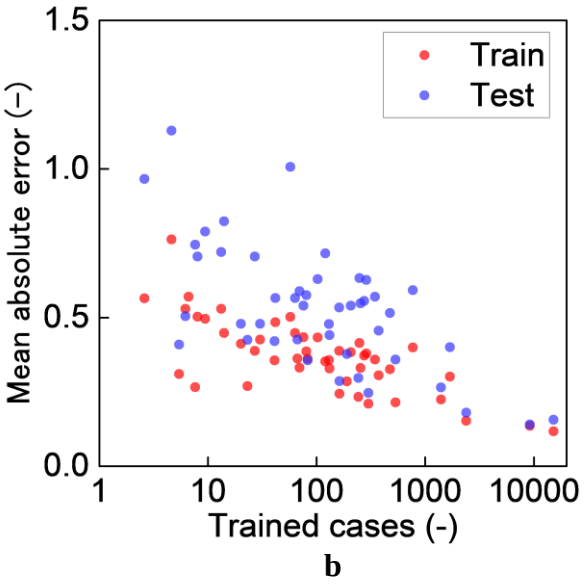

[1] R^2: 0.38
[2] R^2: 0.47
[3] R^2: 0.83
[4] R^2: -50.96
[5] (R^2: 0.68)
[6] R^2: 0.62
[7] R^2: 0.16
[8] R^2: 0.18
[9] (R^2: 0.70)
[10] R^2: -0.61
[11] R^2: 0.61
[12] R^2: 0.54


[13] R^2: 0.57
[14] R^2: 0.51
[15] R^2: 0.52
[16] R^2: 0.91
[17] R^2: 0.36
[18] R^2: 0.85
[19] R^2: 0.31
[20] R^2: -3.04
[21] R^2: 0.50
[22] R^2: 0.22
[23] R^2: 0.62
[24] R^2: -0.27

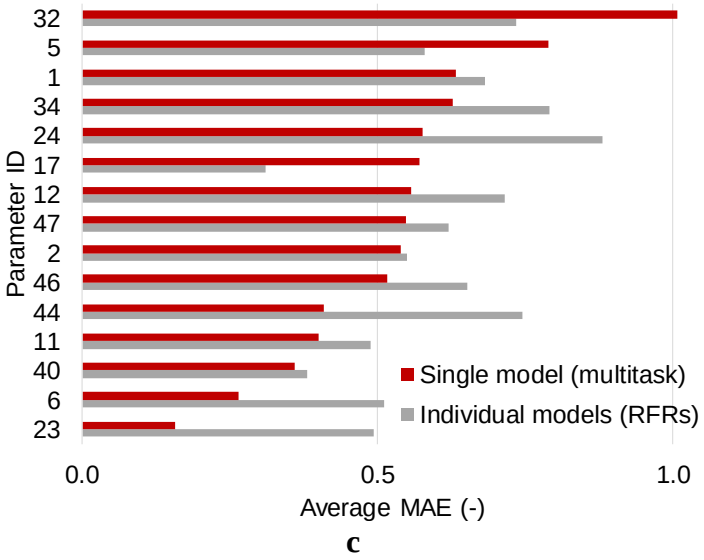

[25] R^2: -1.85
[26] (R^2: 0.71)
[27] (R^2: 0.30)
[28] R^2: -0.73
[29] R^2: -2.48
[30] R^2: 0.52
[31] (R^2: 0.95)
[32] R^2: 0.27
[33] R^2: 0.82
[34] R^2: 0.09
[35] R^2: 0.65
[36] (R^2: 0.48)


[37] (R^2: 0.55)
[38] R^2: 0.00
[39] R^2: -0.31
[40] R^2: 0.16
[41] R^2: -0.07
[42] R^2: -0.16
[43] R^2: 0.22
[44] (R^2: 0.91)
[45] R^2: -2.24
[46] R^2: 0.62
[47] R^2: 0.11

a

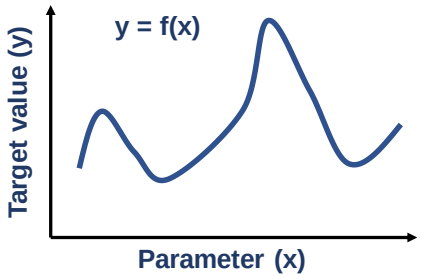


b



c

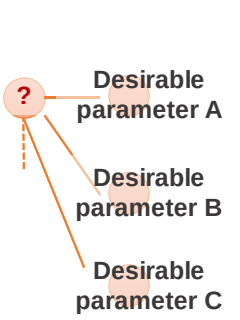
Supplementary Fig. 14. Results for the multitask learning.
a Larger graphs of Fig. 5.
b Relationship between mean absolute error and the number of trained cases (*n*) for each parameter. The graph is plotted with data from Supplementary Table 1.
c Comparison of multitask training and single-task training by random forest regressors (RFRs, see Supplementary methods for protocols). For each parameter, the average MAEs for the test datasets are shown. Parameter IDs can be referred in Supplementary Table 1. The 14 databases and Wikipedia database were selected as the datasets for multitask (Fig. 5) and single-task training. Smaller prediction errors with most parameters by the multitask model can be attributed to 1) synergistic effect of multitask learning and 2) larger amounts of training data inputted to the graph model (i.e., RFRs tend to suffer from the smaller amounts of data because of the limitation of the inputtable types of data).



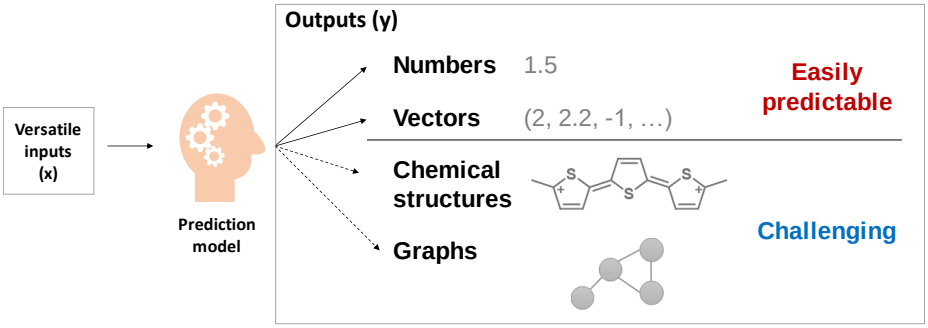
a

	Parameter y_1	Parameter y_2	Parameter y_3	...	Structure
1	1	5	2	...	A
2	8	?	8	...	B
3	?	4	?	...	C
...	...	...	...	...	...

b

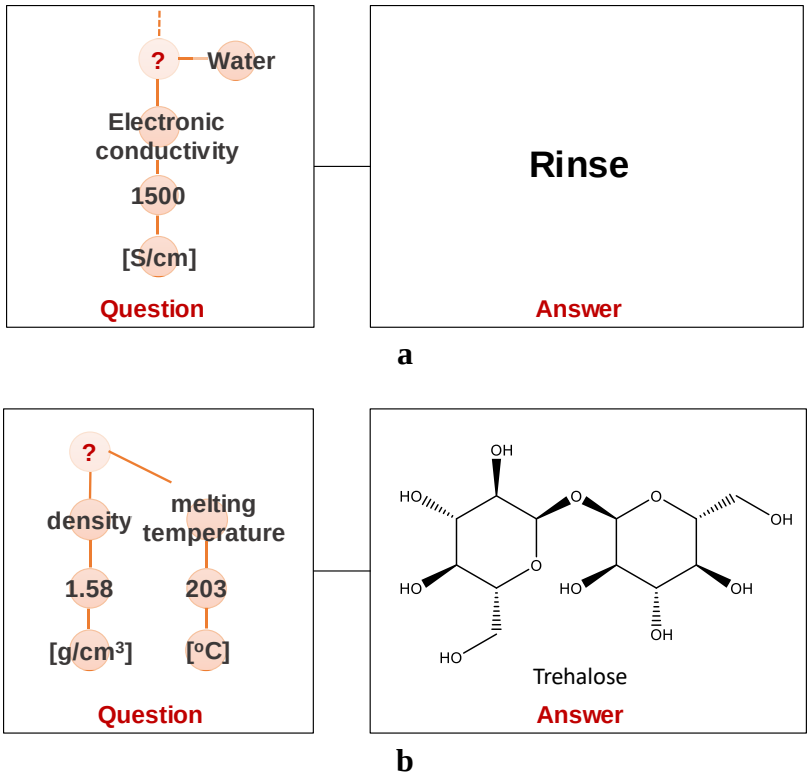


c



d

Supplementary Fig. 15. Challenges in inverse problem-solving.
a Uniqueness of answer problem. Target value (*y*) is determined if its related parameters (*x*) are given (*y* = f(*x*)), but the opposite is not valid with most functions. For example, melting point (*y*) is determined when the chemical structure (*x*) is given, but structures cannot be determined only from melting points because multiple compounds may give the same *y*.
b Table format problem. Adding other desirable parameters (y_2, y_3, ...) is an effective way of obtaining a unique *x* (e.g., predict a structure with a specific melting point, a specific density, or a specific heat capacity). However, preparing the complete table of experimental parameters is normally costly. Typically, there will be many missing values when data is collected from the literature.
c Graph approach for inverse problems. In contrast to tables, flexible graphs are robust against the missing values and easily reusable in other projects. Adding desirable parameters to the question node is sufficient to cope with the uniqueness problem.
d Informatic challenge. Machine learning models, which process all data in digital numbers, can predict numbers and numeric arrays easily. However, complex information, including chemical structures and graphs, is still difficult to generate, even with cutting-edge neural networks.



Supplementary Fig. 16. Actual examples of inverse problems. a Text prediction from the electrical conductivity of PEDOT-PSS. **b** Chemical structure prediction from melting point and density.

Supplementary Tables

Supplementary Table 1. Average R^2 scores and mean absolute errors (MAE) for each parameter after training the 14 databases. Ninety percent of the data are selected randomly as the training datasets. MAE is calculated for the z-scores of the parameters. Random data splitting and training were repeated 8 times to obtain the averages of R^2 and MAE (n =8 independent experiments). Standard deviations are also shown in the table. Full results (including those of training dataset and random forest regressions) are summarized in the Source Data.

ID	Parameter	Results for test dataset				
		R^2		MAE		n
		average	±	average	±	average
1	Absolute molar magnetic susceptibility	0.09	0.25	0.63	0.20	29
2	Absolute standard enthalpy of formation	-0.04	0.46	0.54	0.08	23
3	Amorphous density	0.68	0.12	0.48	0.14	14
4	Amorphous thermal conductivity	-7.38	19.23	0.71	0.47	4
5	Band gap	-4.20	1.33	0.79	0.18	2
6	Boiling temperature	0.72	0.10	0.27	0.04	153
7	Converted redox potential (V vs NHE)*	0.19	0.45	0.43	0.12	7
8	Crystalline density	0.45	0.37	0.44	0.21	13
9	Crystalline thermal conductivity	-0.70	0.00	0.51	0.39	1
10	Decomposition temperature	-1.02	1.83	0.63	0.35	11
11	Density	0.49	0.07	0.40	0.03	180
12	Dipole moment	0.42	0.15	0.56	0.05	33
13	Electric conductivity	0.44	0.27	0.25	0.05	35
14	Flash temperature	0.54	0.12	0.46	0.09	44
15	Glass expansivity	0.35	0.43	0.57	0.23	6
16	Glass transition temperature	0.90	0.03	0.18	0.02	266
17	Heat capacity	0.41	0.14	0.57	0.08	41
18	Ionic conductivity	0.83	0.04	0.14	0.02	1011
19	Ionization energy	0.34	0.06	0.59	0.03	87
20	Liquid expansivity	-0.30	1.29	0.54	0.18	8
21	Liquid heat capacity	-0.06	1.10	0.48	0.17	4
22	Melting enthalpy	-2.16	3.94	0.59	0.37	9
23	Melting temperature	0.48	0.39	0.16	0.01	1697
24	Molar heat capacity	-0.29	0.86	0.58	0.17	9
25	Molar volume	-0.80	1.14	0.29	0.15	17
26	Oxidation potential (V vs Ag/Ag^+)	0.03	0.70	0.71	0.46	2
27	Oxidation potential (V vs Fc/Fc^+)	0.00	0.00	1.13	0.87	1
28	Oxidation potential (V vs SCE)	0.16	0.51	0.48	0.17	3
29	Oxidation potential (V vs SHE)	-1.58	1.27	0.82	0.93	2
30	Oxidation/reduction potential (V vs Li/Li^+)	0.73	0.16	0.36	0.11	9
31	Oxidation/reduction potential (V vs SHE)	-33.99	0.00	0.75	0.18	2
32	Partition coefficient	-1.31	1.62	1.01	0.28	6

33	Permittivity	0.76	0.13	0.30	0.12	27
34	pKa	0.00	0.18	0.63	0.20	29
35	Polarizability	0.39	0.54	0.38	0.06	20
36	Reduction potential (V vs Ag/Ag^+)	0.00	0.00	2.21	0.77	1
37	Reduction potential (V vs NHE)	-1.26	0.00	0.97	0.04	2
38	Reduction potential (V vs SCE)	0.33	1.10	0.43	0.17	3
39	Reduction potential (V vs SHE)	-14.03	27.89	0.72	0.51	2
40	Refractive index	0.17	0.20	0.36	0.17	58
41	Solid heat capacity	-0.57	1.11	0.57	0.20	4
42	Solubility parameter	-0.17	0.29	0.72	0.19	13
43	Surface tension	0.36	0.25	0.53	0.18	18
44	Thermal conductivity	-10.70	16.07	0.41	0.19	2
45	UV cutoff	0.39	1.06	0.42	0.11	5
46	Vapor pressure	0.55	0.09	0.52	0.07	50
47	Viscosity	0.21	0.24	0.55	0.07	30

*In addition to the reported redox potentials (e.g., vs. Li/Li^+), converted potentials against standard electrodes were estimated and included in the databases.

Supplementary Table 2. Performance of the model trained with the inverse problems[a)]

Target[b)]	**Training dataset**	**Test dataset**
Numbers (R^2 score)	0.84	0.85
Text (accuracy)	0.96	0.96
Compound (accuracy)	0.35	0.35

a) Each node in all graphs of the 14 databases are prepared as problems. Ninety percent of the data is used as the training set. b) Model is trained with the 64-dimensional output to reduce the mean square errors of the 60-dimensional context vectors. R^2 score for numeric values are calculated from the averages of the 60-dimensional vectors and the experimental values. For predicting text and compounds, the candidates giving the largest cosine similarity with the predicted vectors are selected from the databases.

Supplementary Table 3. Typical mistakes made by the prediction model.

Answer	**Prediction**
Ionic conductivity	Electrical conductivity
Melting temperature	Boiling temperature
Random	Block
Melting point	Flash temperature
Absolute standard enthalpy of formation	Melting enthalpy
Vapor pressure	Solubility parameter
Ionic conductivity	Electrical conductivity

Supplementary Discussion

Comparison of our approach with traditional methods (Supplementary Fig. 2)

(1) Language models

Texts play essential roles in describing scientific information in materials informatics. For instance, experimental procedures (e.g., preparation of PEDOT-PSS films) are generally explained by texts. They are often treated by language models, such as LSTM[2] and BERT[3], which can input the variable-length data of words. Even compound information can be treated by the language models after acquiring the "meanings" of the compounds by appropriately data mining (i.e., prepare embeddings for compounds, expressed by words).[4] The approach is appreciated to cope with the big data of science because most available information is recorded by texts.[4]

On the other hand, the main limitations of the language models for compounds should be the high cost for machine learning and weakness against new cases. Orthographical variants are often detected in scientific papers (e.g., H_2O, water, and dihydrogen monoxide represent one specific compound). Language models need vast amounts of data to understand the commonality of different expressions. The worst (but often observed) cases for the models would be the newly reported, but only nicknamed compounds in the manuscripts (e.g., compound 1, polymer 1, and P1). Even conventional compounds have uniqueness problems. For instance, there are many types of "polyethylene"s with different density, molecular weight, and crystalline structures. Therefore, appropriate embedding may not be easy for new and complex chemicals.

(2) Normal models

More accurate ways of expressing compound information are using special algorithms designed for chemicals (e.g., fingerprints and descriptors)[5-7] to outputting their fixed-dimensional vectors. Even features of new compounds can be calculated accurately in most cases. Conventional machine learning models (e.g., multi layer perceptron, support vector regressors, and random forest regressors) can be used to input the fixed-dimensional vectors and to predict the properties of the corresponding chemicals.[8] This is a typically employed approach in materials informatics.[9]

The limitation of the approach is the inflexibility of machine learning. Since most models can accept only the fixed-dimensional vectors, complex information, such as texts, are difficult to be interpreted (multitask training is also difficult, as discussed in the main manuscript).[2,3]

(3) Our approach

To combine both text and compound information, we introduced graph structures. On each node, feature vectors of corresponding words, compounds, and values were embedded. Word and compound embeddings were precisely made by a pretrained model of BERT and a concept of fingerprint, respectively. Graphs were processed by graph neural networks, outputting fixed-dimensional vectors (which can be treated by conventional machine learning models). The combination of the flexible graphs and accurate embedding is a key to achieve the concepts of process informatics, multitask training, and solving inverse problems.

References

1 McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. doi:arXiv:1802.03426 (2018).

2 Sak, H., Senior, A. & Beaufays, F. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. doi:arXiv:1402.1128 (2014).

3 Devlin, J., Chang, M.-W., Lee, K. & Toutanov, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. doi:arXiv:1810.04805 (2019).

4 Tshitoyan, V. *et al.* Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95-98, doi:10.1038/s41586-019-1335-8 (2019).

5 Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742-754, doi:10.1021/ci100050t (2010).

6 Moriwaki, H., Tian, Y. S., Kawashita, N. & Takagi, T. Mordred: a molecular descriptor calculator. *J. Cheminform.* **10**, 4, doi:10.1186/s13321-018-0258-y (2018).

7 Gomez-Bombarelli, R. *et al.* Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **4**, 268-276, doi:10.1021/acscentsci.7b00572 (2018).

8 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, doi:10.1021/jacs.9b11442 (2020).

9 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, doi:10.1038/s41524-017-0056-5 (2017).