Supplementary Information

Analyzing Machine Learning Models to Accelerate Generation of Fundamental Materials Insights

Mitsutaro Umehara[1,2], Helge S. Stein[1], Dan Guevarra[1], Paul F. Newhouse[1], David A. Boyd[1], John M. Gregoire[1,]*

[1] Joint Center for Artificial Photosynthesis, California Institute of Technology, Pasadena CA 91125, USA. E-mail: gregoire@caltech.edu

[2] Future Mobility Research Department, Toyota Research Institute of North America, Ann Arbor, MI 48105, USA.

[*] Email: gregoire@caltech.edu

1 CNN model selection compared to other models

The structure candidates are first selected by the score of prediction. If the score is the same level, then the simpler model should be better; for example, a model with two CNN layers for spectrum input has similar prediction score but eliminated by this criterion. However, too simple model could not lead the appropriate physical interpretation; for example, a model with only one neural network layer for composition input (no (f) layer in Figure 2) can distinguish rare-earth metals from transition metals but cannot distinguish rare-earth metals from Bi. We also carried out 5-fold cross validation of the CNN model and obtained R^2 score of 0.764, which is slightly better than kernel ridge regression (0.732) and random forest regression (0.741), demonstrating the predictability and ensuring the appropriateness of the CNN model.

2 Gradient in composition space as additional mode of analysis

Gradient direction in composition space obtained by gradient analysis[1-4] shows a guidance to obtain higher performance. Figure S1 shows the gradient direction as arrows in several materials subspace in {Bi, V, Mo, W, Tb, Gd, Dy} with different Bi/V ratios.

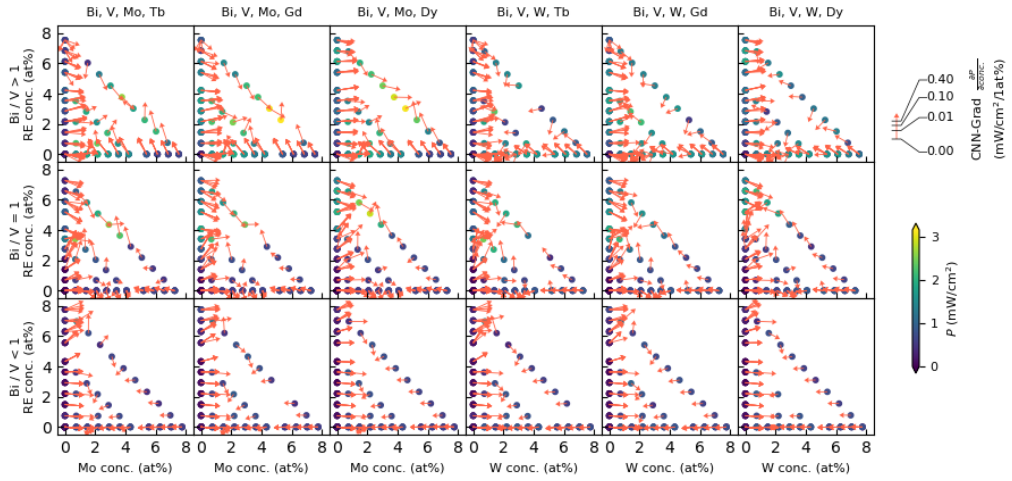

Figure S1. Power generation performance P mapping in composition space with its gradients (arrows). Color of circles indicate the power generation performance. Arrows indicate the direction of gradient and its strength (The displayed scale of gradient is not linear; [length of arrow] = [strength of gradient]$^{0.2}$).

3 Weights of the first layer for elemental data in neural network model

The weights of first neural network layer (e) of 8 trainings are shown in FigureS2. The dimension of the matrix is 7 (input) × 16 (units).

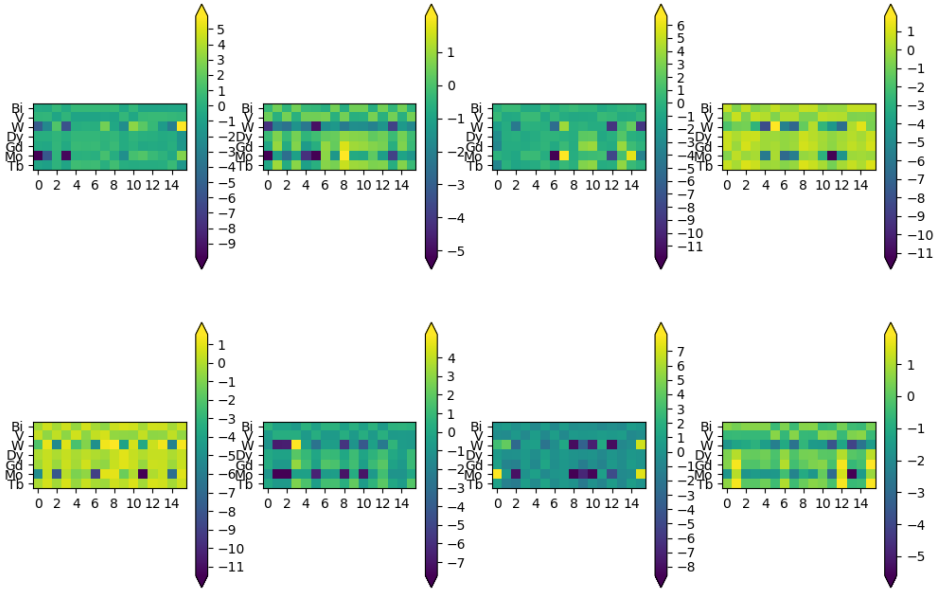

Figure S2. Heat map of weights of the first layer for elemental data in neural network model in 8 training runs. There is an arbitrariness for permutation of the units. Weights of W and Mo are similar and those of Gd, Tb, and Dy are similar in a run.

4 A note for training concatenated neural network

The model used in this study is a concatenated model of spectrum data model and element data model as shown in Figure 2. We train this model in three steps. First, we ignored the elemental data layers (d, e, and f) and trained using only spectrum data input. The initial value for weights of all layers is randomly chosen. Second, we ignored the spectrum data layers (a, b, and c) and trained using only element data input. The initial value for weights of all layers is also randomly chosen. Finally, we trained concatenated model using pre-trained weights of layer (b, c, e, and f) in the first and second step. Weights of layer (h, i, and j) is also randomly chosen in the third step. In this way, we could obtain better prediction performance compared with training the whole model at once.

We repeated these processes 8 times to obtain average and standard deviation of gradient, with the moderate variations in the model with different random initializations demonstrating that each model converges to a local minimum of the loss function, resulting in a predictive model that captures the primary data relationships.

5 A note for the number of calculations

We calculated models for 8 times and averaged their gradients to ensure the reliability of calculation, in particular insensitivity of the results to random seeding. Figure S3 and S4 shows the same figures as Figure 3 and 6, respectively, but averaged over 24 models, which we additionally calculated. The result of this 24-model calculation agreed with the result in the main text. Reader can check the result using our code on author's GitHub[5]. Slight variations in the calculated gradients (generally within the uncertainty illustrated in the figures) may be obtained due to random initialization of weights in neural network, randomly shuffled order of dataset in one batch, and GPU's error correction quality. Our GPU system is NVIDIA GeForce 940M.

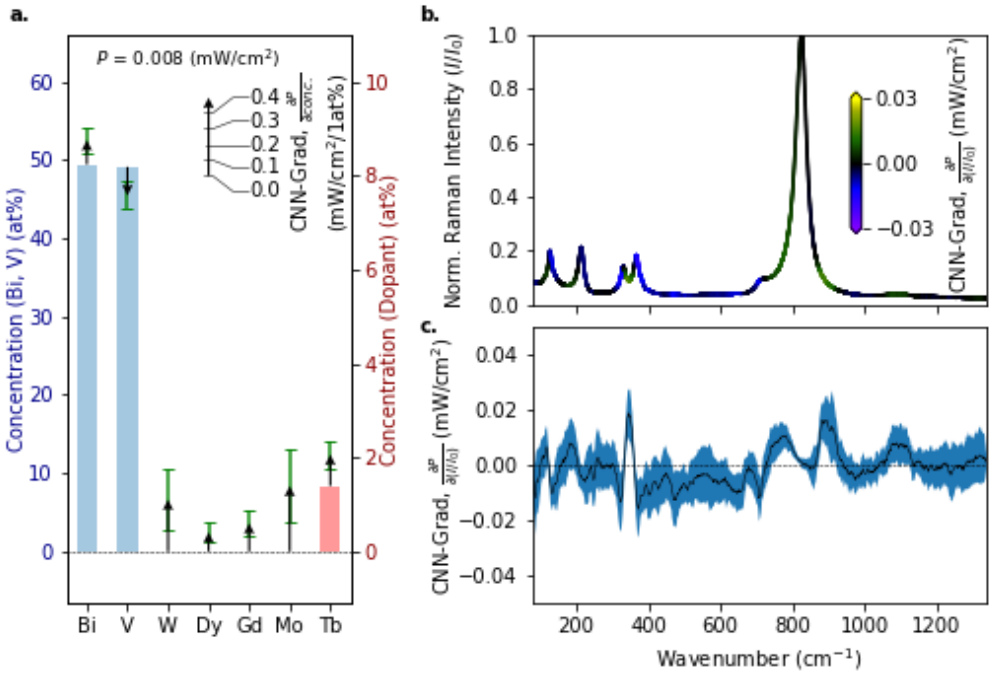

Figure S3. The same figure as Figure 3 but using the result averaged over 24 calculations.

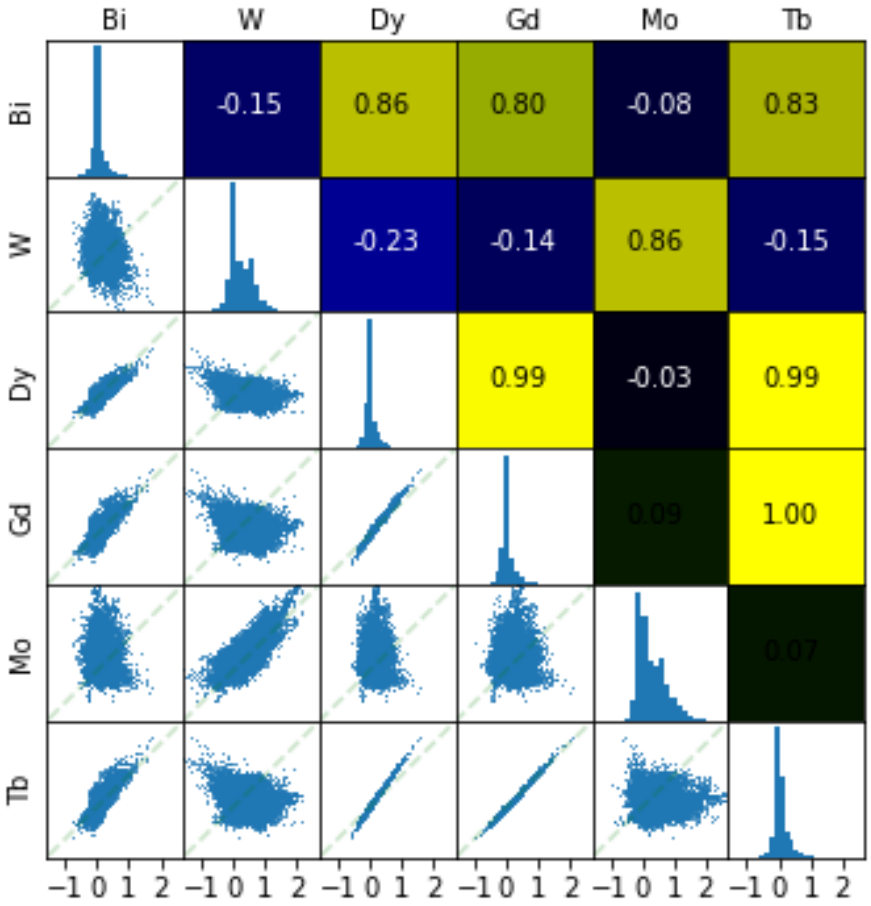

Figure S4. The same figure as Figure 6 but using the result averaged over 24 calculations.

Reference

1. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. Preprint at http://arxiv.org/abs/1312.6034 (2013).
2. Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision* 818–833 (2014)
3. Springenberg, J. T., Dosovitskiy, A., Brox, T. & Riedmiller, M. Striving for Simplicity: The All

Convolutional Net. Preprint at http://arxiv.org/abs/1412.6806 (2014).

4. Chollet, F. How convolutional neural networks see the world. https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html. (2016)
5. Available at https://github.com/johnmgregoire/CNN_Gradient_Analysis