Neural networks for a quick access to a digital twin of scanning physical properties measurements

Kensei Terashima,1,∗ Pedro Baptista de Castro,1,2 Miren Garbiñe Esparza Echevarria,1,2 Ryo Matsumoto,1 Takafumi D Yamamoto,1
Akiko T Saito,1 Hiroyuki Takeya,1 and Yoshihiko Takano1,2

1National Institute for Materials Science, 1-2-1 Sengen, Tsukuba, Japan
2University of Tsukuba, 1-1-1 Tennoidai, Tsukuba, Japan
Abstract

For performing successful measurements within limited experimental time, efficient use of preliminary data plays a crucial role. This work shows that a simple feedforward type neural networks approach for learning preliminary experimental data can provide quick access to simulate the experiment within the learned range. The approach is especially beneficial for physical properties measurements with scanning on multiple axes, where derivative or integration of data are required to obtain the objective quantity. Due to its simplicity, the learning process is fast enough for the users to perform learning and simulation on-the-fly by using a combination of open-source optimization techniques and deep-learning libraries. Here such a tool for augmenting the experimental data is proposed, aiming to help researchers to decide the most suitable experimental conditions before performing costly experiments in real. Furthermore, this tool can also be used from the perspective of taking advantage of reutilizing and repurposing previously published data, accelerating data-driven exploration of functional materials.

I. INTRODUCTION

The recent global rise of databases on materials as well as accessible repositories and advanced text mining techniques has led to the evolution of materials informatics as an effective tool for exploration and design of functional materials [1]. Along with the increase of the available crystal structural input data, the emergence of high-throughput first-principles calculation systems has enhanced a variety of materials estimated property data useful for statistical analysis [2–4]. In addition, the application of machine-learning methods has proved to be quite efficient for dealing with a large amount of materials data [1, 5], allowing the extraction of useful information that has been hidden or too complicated for an ordinal human to percept. As a result of the recent rapid growth of such tools for handling materials data, the process for obtaining candidate materials in cyberspace has been significantly accelerated, while the bottleneck for the discovery of functional materials remains on the actual synthesis and experimental evaluation in real space [6]. This is because the cost for obtaining experimental data is usually high, especially when researchers have to use a state-of-the-art shared facility with fixed machine time. Furthermore, not only

* TERASHIMA.Kensei@nims.go.jp
the experiment itself but also its preparation can be time-consuming as sometimes one has to start from optimization of the sample synthesis process, a typical aspect for physics and materials science measurements.

Such a costly-in-real situation is even more serious in industrial fields, where a large amount of money has to be spent for the production of "real" items. To overcome this issue, there an idea called “digital twin” was produced, where one builds a simulation model that takes account of the data in real space and tries to optimize the parameters in the cyber space before real synthesis [7]. For being beneficial, the model has to correspond enough to the objective, and the cost for the simulation both in money and time has to be lower than the actual production. Here again, the application of machine-learning methods has greatly reduced the duration required for the execution of simulation as compared to that for performing conventional simulation algorithms. Recently, the use cases of such a digital-twin approach aided by machine learning are becoming popular also in academic research. For instance, machine-learning was used for approximating time-consuming simulation results for fluid dynamics, allowing researchers to perform a quick optimization of the growing condition of objective crystals [8].

In this context, such a digital-twin approach is expected to be effective and highly useful also for physics and materials science measurements, where the objective properties are sometimes deduced from experimental data by applying mathematical operations such as derivatives or integration. Especially in physics, the development of new approaches or ideas to data analysis, creates an opportunity to revisit or reanalyze previously published data, allowing researchers to draw new insights from this previous knowledge. However, it is quite often that the data are non-evenly spaced along multiple axes due to experimental confinements, and it may contain non-linear responses that cannot be interpolated easily. In such a case, it is quite difficult to estimate the target quantity from preliminary data. On the other hand, artificial neural networks are known to have high flexibility and adaptivity in principle [9, 10]. It has been also discussed [11] that neural networks learning can be as efficient as other approximation methods from a viewpoint of minimizing loss using the number of parameters as low as possible. They have proved to be even useful when the objective has inhomogeneity in smoothness [12], or when the objective has several discontinuities [13].
Thus, approximation using this method is expected to be suitable for materials science and physics experimental data that tend to contain discrete change due to first-order transition and/or multiple peak structures.

In this article, we show that learning preliminary data of materials science by neural networks provides us a way to quickly build a model that can describe the target property dependencies on scanning axes (treated as features). In order to ensure the accuracy of the model that can be built on-the-fly during the experiment, several open python libraries have been used in combination, as well as several tools for efficient learning. Once the model is built, the simulation within the learned feature range can be done instantly, which enables researchers to evaluate the experimental plan as well as its cost (Figure 1(a)). We propose that such an approach of data augmentation helps researchers to use preliminary- or deposited data effectively for performing the data-driven search of functional materials.

![Graphical concept for application of digital-twin approach to materials property measurements for performing efficient experiments using preliminary data as train data.](image)

**FIG. 1.** (a) Graphical concept for application of digital-twin approach to materials property measurements for performing efficient experiments using preliminary data as train data. (b) Combination of open libraries used in this study to quickly build a fully-connected feedforward neural networks model for simulating experimental data.

### II. CONSTRUCTION OF NEURAL NETWORKS MODEL

In our approach, the constructed code relies mainly on 4 open libraries available to the public in addition to the Anaconda package [14], shown in Figure 1(b). As a machine-learning body, we built fully-connected feedforward neural networks, using Keras with
Tensorflow backend (v.2.x) [15, 16]. Keras has several optimizers by default, and among them, we tried stochastic gradient descent, Nesterov accelerated gradient [17], root mean square propagation [18], and adaptive moment estimation (Adam) [19] for several times to our data shown in Section 3. For our data, Adam tended to be the most stable and fast for minimizing loss of mean squared error between predicted values and training data (not shown). We set the number of layers, nodes, learning rate, and batch size as hyperparameters to be optimized. Once those hyperparameters are suggested, random weights and biases are given as initial values in Keras and they are tuned by the optimizer as the epoch proceeds in a single training run. Thus the learned result in a single run varies by the initial random values, even though there could be a general trend for obtaining a moderate set of hyperparameters for example the case in learning rate and batch size [20–22]. To be quick, we confined the search space of the hyperparameters in the following range: $2 \leq \text{number of layers} \leq 10$, $50 \leq \text{number of nodes} \leq 200$, $1e^{-5} \leq \text{learning rate} \leq 1e^{-4}$, and $16 \leq \text{batch size} \leq 1024$ varying on the data size. To obtain the best possible learned result in a fixed time, we used the Bayesian optimization package Optuna [23]. By combining a relational database provided by MySQL [24], Optuna receives each training result with a set of hyperparameters running in parallel, so that it can determine a set of hyperparameters for the next run based on Tree Parzen Estimator [25]. Typically, the best set of hyperparameters is found by at most $\sim$30 runs. After many trials, we found that overfitting is welcome for our purpose to learn the detail of the real experimental data (including noise level and characteristics of experimental machines), and that cross-validation simply increases the required time for training. Thus we used the whole data for training. For visualization of data and controlling multiple Optuna runs in parallel, we used the Streamlit library [26] to construct a graphical user interface (GUI) that works on the web browser.

To obtain the best model during the hyperparameter search, we used three functions equipped in the above libraries. First, is early stopping implemented in Keras, with the option of saving the best model in the run, where the maximum epoch number is 500. The second is the learning rate scheduler implemented in Keras, where we change the learning rate to be 10 times smaller value from its initial value after 100 epochs. The third is pruning with the Asynchronous Successive Halving algorithm [27] implemented in Optuna, where it examines if each run shall be pruned based on the comparison of current score with those
of running parallel by $25^n$ epochs where $n$ is an integer greater than or equal to zero.

The approach used here might be primitive as compared to recent state-of-the-art combinations of neural networks architectures, optimizers, and other techniques, but we tried to keep the system simple, fast, and easy to use as much as possible, considering that the target users are not specialist of data science but materials scientists like us. As a result, we could quickly finish learning small datasets typical for physical property measurements (number of data $\sim$ a few thousand) shown in the next section, even without using graphics processing units (GPUs). More detail on case studies for the duration and used computers are in Supporting Information. A tutorial for how to get ready and perform the neural networks learning and simulation shown in this paper will be available at https://www.github.com/kensei-te/mat_interp.

## III. RESULTS AND DISCUSSION

In this section, we show 4 typical use cases as an example. The first is the magnetization curve of a well-known magnetocaloric material ErCo$_2$, where 1$^{st}$ order transition occurs that accompanies a steep change in the magnetization as a function of temperature and the transition temperature migrates by application of an external magnetic field. The second example is the magnetization curve and estimation of magnetocaloric effect in another magnet Fe$_3$Ga$_4$ that shows 1$^{st}$ order transition but its transition temperature changes more drastically by applied field than in the first example of ErCo$_2$. The third example is the resistance data of the PdH superconductor which is relatively noisy as compared to above examples. The fourth example is a mapping data of angle-resolved photoelectron spectroscopy (ARPES) intensity on Fermi energy of La(O,F)BiS$_2$ superconductor taken above superconducting temperature ($T_c$) as a demonstration of the application of the method for two-dimensional intensity map, consisting of a number of peak structures.
A. \( M(T, H) \) of ErCo\textsubscript{2} magnet and estimation of its magnetocaloric effect

First, we show a case for physical property data that have 1\textsuperscript{st} order transition, namely a steep change in the observed value. ErCo\textsubscript{2} is known to be a ferrimagnet with its ordering temperature \( T_{\text{Curie}} \) of \( \sim 35 \) K \[28\]. It exhibits a 1\textsuperscript{st} order magnetic transition accompanied by magnetostriction, and the material is one of the most popular materials for magnetic refrigeration \[29–31\]. To evaluate the magnitude of magnetic entropy change |\( \Delta S_M \)|, one of the most common ways \[30\] is to measure the magnetization of the sample as a function of temperature and applied magnetic field change \( \Delta H \) (from 0 to \( H \)) either isothermal or isofield, and deduce |\( \Delta S_M \)| by applying following Maxwell’s equation for the observed magnetization data:

\[
\Delta S_M(T, \Delta H) = \mu_0 \int_0^H \frac{\partial M(T, H')}{\partial T} dH'
\]

The equation includes an integration along the applied field \( (H) \). Practically, with such discrete experimental data, trapezoidal integration can be applied for a numerical integration, which corresponds to a linear interpolation along the \( H \)-axis. However, if we estimate |\( \Delta S_M \)| in this manner from the experimental data shown in Figure 2(a) taken by a coarse magnetic field step, it ends up with extrinsic oscillation of estimated |\( \Delta S_M \)| values as shown as a gray solid line in Figure 2(e). This is simply because of the failure of linear interpolation for data having 1\textsuperscript{st} order transition (see Supporting Information for technical details). To avoid this extrinsic effect for estimation of the target property, one has to use a finer magnetic field step, so that linear interpolation starts to work well since \( \partial M/\partial T \) between two steps has a finite overlap each other. This means one has to know the proper measurement step before the measurement itself. In such a case, neural networks simulation based on preliminary data works efficiently as it can perform a non-linear interpolation. Figure 2(b) shows the predicted magnetization values by a constructed model for the conditions in the train data of Figure 2(a), which corresponds well to the train data, indicating that a plausible model has been made. Once such a model is constructed, the model can perform quick predictions for any desired feature values within the learned range. After a couple of trials, we found in Figure 2(f) that extrinsic oscillation in the simulation is expected to be significantly suppressed when the magnetic field step of the measurement is smaller than 0.2 T. We measured the magnetization in the proposed step as in Figure 2(c), and found
in Figure 2(e) that the experimentally evaluated $|\Delta S_M|$ corresponds very well to what has been predicted by the simulated model.
B. \( M(T, H) \) of Fe\(_3\)Ga\(_4\) magnet and estimation of its magnetocaloric effect

Here we show the case of another magnet Fe\(_3\)Ga\(_4\) and evaluation of its magnetocaloric property, to show the possibility of applying neural networks study for analyzing literature data. The magnetic property and magnetization as a function of temperature and magnetic field have been reported [32], while \( |\Delta S_M| \) of this material has not been reported yet. Figure 3(a) shows our preliminary experimental data that corresponds well to the literature data [32], but both of them do not have fine enough experimental steps to evaluate \( |\Delta S_M| \) of this material. As shown in Figure 3(d) and 3(f), we have performed a simulation of expected \( |\Delta S_M| \) taken on each field step case and found that extrinsic effect for \( |\Delta S_M| \) would not be observed when we measure magnetization by a magnetic field step less than 0.1 T for this material. Considering the required experimental time, we chose a 0.075 T step, and the simulated data showed an excellent correspondence with the real experimental data as shown in Figure 3(c)-(f). The simulation also tells us that, the estimated \( |\Delta S_M| \) is at most 0.25 J kg\(^{-1}\) K\(^{-1}\) by a field change of 0-1 T (which is small as compared to those of other magnetocaloric materials [30]) despite this measurement covers wide temperature range of 10-200 K with fine magnetic field step, hence it is expected to take approximately 1 week. Therefore if one is simply looking for a material with high \( |\Delta S_M| \), the simulation tells us that this material may not be suitable to be synthesized and measured by spending a fair amount of time. In other words, such a simulation is not only useful for analyzing the past preliminary or repository data from a different point of view, but also it can help researchers to evaluate the cost of performing the experiment by leveraging the confined knowledge available from preliminary- or literature data.

C. \( R(T, H) \) of PdH superconductor

Next, we show the case for data with a lower signal-to-noise ratio. For this purpose, we used a part of our resistance data of PdH [33] as train data (Figure 4(a)). As can be seen in Figure 4(b), the model does not necessarily capture the ultrafine structure coming from noise, however it grabs not only the overall tendency of the lineshape but also several fine structures that might be influenced by noise. The inability of tracking the full noise is because of the fact that the number of layers and nodes in the model is limited and lower.
FIG. 3. (a) Preliminary magnetization data of Fe$_3$Ga$_4$, which is used as the train data for construction of model. (b) Simulated magnetization by the trained model for the conditions in the train data. Gray solid lines show the train data. (c) Magnetization of Fe$_3$Ga$_4$ with magnetic field step proposed by the model. (d) Simulated magnetization by the trained model. (e) Estimated $|\Delta S_M|$ of Fe$_3$Ga$_4$ for a field change of 0-1 T. (f) Simulated $|\Delta S_M|$ for different experimental steps of magnetic fields. Gray solid line shows the estimated $|\Delta S_M|$ from (a) by applying trapezoidal integration along magnetic field direction. Blue arrows between figures indicate an order of workflow of the measurements and simulations.
than the number of total train data points. Indeed, neural networks are sometimes used
for denoising purposes [34, 35]. On the other hand, our intention here is to simulate the
experiment as it is, including the extrinsic effect such as noise and unique characteristics of
the measurement apparatus, based on the non-evenly spaced preliminary data. Comparing
the experimental data in Figure 4(c) with simulated data in Figure 4(d), it can be
said that the model in the current approach can tell us how the expected lineshapes can be
affected by the noise level appeared in the train data, with a finite inevitable denoising effect.

![Picture](image)

**FIG. 4.** (a) Train data of resistivity measurements for PdH, taken from a part of the whole experimental data in (c). (b) Simulated resistance by the trained model for the conditions in the train data. Gray solid lines show the train data. (c) Experimental data of resistivity measurements for PdH superconductor [33]. (d) Simulated resistance by the trained model for the same conditions in (c). Gray open circles show the conditions seen by the model during training.

**D. Angle-resolved photoelectron spectroscopy intensity map of La(O,F)BiS$_2$**

As a last example, we show that our quick and simple approach is also applicable to two-
dimensional intensity map, namely, ARPES intensity map at Fermi energy in La(O,F)BiS$_2$
superconductor on normal state (above $T_c$). ARPES intensity at fixed energy tends to be
high in angular area where corresponding electronic state exists, thus the data consists of a
number of peak structures (see Supporting Information for lineshape of each cut). In Figure 5, the ARPES intensity is shown as a function of two-dimentional angles with respect to the normal angle to the sample surface, that corresponds to two-dimentional wave vectors in reciprocal space [36]. For constructing such a mapping data, it is quite often that one measurement is performed with fixing one of the axis (for instance $\theta_2$ in the figure) and keep concatenating data with different angle of $\theta_2$. Therefore a preliminary measurement with coarse angle step of $\theta_2$ (Figure 5(a)) can be performed before a serious measurement with fine angular steps and better statistics (Figure 5(c)), so that one can determine the objective measurement area of angles. With help of neural networks learning, we can simulate how data would become with specific measurement area and steps as in Figure 5(d), thus the simulation helps researchers to make a decision for planning the experiment. Here we note that the simulated pattern captures characteristics of the measurement apparatus, namely high sensitivity at edge and detection limit in certain angle, so it is suitable for estimating the effect of such instrumental conditions as well. However, the simulated pattern is affected by the presence of sizable noise, hence the pre-processing of train data with other noise reduction method might help further though such process may take longer time. Alternatively, it is also possible to take an avarage of predictions by several models found during learning process. We also note that although our simple approach is applicable on-the-fly to such a mapping data including non-evenly spaced ones, other sophisticated methods such as super resolution convolutional neural networks-based approach [37–39] would perform certainly better especially for evenly spaced data such as photographs if one can spare enough time for training and has an access to powerful GPUs.

E. Scope and limitation of current approach

Here we note the applicability/inability of our approach. The code is available at https://www.github.com/kensei-te/mat_interp including how to get ready to use GUI, which runs in Linux and Mac OS (see Supporting Information for details). It can learn quickly the data so that one can perform the simulation during the experiment, especially when the total number of data points kept up to a few tens of thousands. For this reason, it is preferred to keep the dimension of data (i.e. number of feature columns) low, even though it can deal with multidimensional data. The approach would be useful mainly for two
FIG. 5. (a) Roughly-scanned ARPES intensity distribution as a function of angles with respect to sample normal, used as train data. (b) Simulated ARPES intensity map by the trained model for the conditions in the train data. Green dotted square shows the simulated angular area in (d). (c) Same as (a) but with proposed angular steps by simulation in (d). (d) Simulated ARPES intensity map by the trained model for estimation of measurement conditions in (c).

cases: (i) Simulation of a costly experiment from preliminary experimental data: based on the roughly-scanned data, one can simulate how the data and resultant estimated target property would be, when taken in fixed experimental steps and ranges. Thus it helps researchers to plan and perform efficient experiments. This is beneficial especially when the target properties are deduced from mathematical operations of experimental data and hence are depending on experimental conditions, as shown in Section 3.1 and 3.2. (ii) Aligning and augmentation of repository data: there is an increasing number of accessible materials data sets, that have been published in the past. It is quite often that one encounters such problems that despite the material being of interest, there is a mismatch or lack in the experimental conditions as compared to what is required in order to perform additional analysis. As the current method can adapt readily to non-evenly spaced data with a non-linear response, it will help researchers to judge whether the material is worthy to be examined further by reusing this data for the simulation method proposed here, to spare a certain amount of experimental cost as shown in Section 3.2.
On the other hand, the method is ultimately a mere interpolation, thus the signal that is skipped in the train data with too coarse step will never be predicted in the simulation. For the same reason, though the simulation helps researchers to plan the experiment, the simulated results and estimated values of physical properties themselves should be clearly distinguished from experimental data and be taken with care since it is not certain what will come out in reality until verified by experiment. Also, the prediction by learned models includes how the data will be affected by unique characteristics of experimental apparatus (sensitivity, detection limit, and so on) and noise, as has been shown in Section 3.3 and 3.4. The effect of noise could be examined by comparing several simulated results in different runs (see Supporting Information for details). If the researcher aims to suppress the effect of noise rather than simulating it, other noise reduction methods may work better.

IV. CONCLUSION

In this work, we present a simple and quick method to simulate experimental data by learning preliminary data using fully-connected feedforward neural networks. The approach is shown to be suitable for materials science experimental data with typical examples. Such a tool would help researchers to deal with preliminary- or past repository data efficiently, supporting decision-making during the experimental research. A tutorial on how to get ready and perform the neural networks learning and simulation shown in this paper will be available at https://www.github.com/kensei-te/mat_interp.

V. EXPERIMENTAL SECTION

A. Sample Synthesis and Preparation

ErCo$_2$ sample was obtained by arc-melting of Er and Co rocks in a vacuum chamber of Ar atomosphere, followed by a 21 hour annealing on 850 °C and quenching by water. Fe$_3$Ga$_4$ sample was obtained by arc-melting of Fe and Ga chunks. The synthesis procedure of PdH sample is provided in [33], while the synthesis procedure of La(O,F)BiS$_2$ sample is in [40].
B. Experimental Conditions

Magnetization of samples has been measured using Magnetic Properties Measurement System (Quantum Design), in field cooling process. Resistance of sample under 2.6 GPa has been measured by four probe method using diamond anvil cell with boron-doped diamond electrodes [33] and Physical Properties Measurement System (Quantum Design). ARPES measurements have been performed at BL28A [41] in Photon Factory KEK, Japan (proposal number 2021G686), with DA30 analyzer (Scienta Omicron) at 20 K (estimated \(T_c\) of the sample is approximately 2.2 K [42]), \(h\nu = 70\) eV.

VI. ACKNOWLEDGEMENTS

The ARPES measurements were performed at BL28A of Photon Factory, KEK (proposal No. 2021G686). The authors acknowledge Wei-Sheng Wang and Taku Tou for testing the installation of the code. This work was supported by the JST-Mirai Program (Grant No. JPMJMI18A3), JSPS Bilateral Program (JPJSBP120214602), and JSPS KAKENHI (Grant Nos. 20K05070, 19H02177). P.B.C. and M.G.E.E. acknowledges the scholarship support from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

[1] V. Stanev, K. Choudhary, A. Kusne, J. Paglione, and I. Takeuchi, Artificial intelligence for search and discovery of quantum materials, Communications Materials 2, 105 (2021).
[2] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, The Materials Project: A materials genome approach to accelerating materials innovation, APL Materials 1, 011002 (2013).
[3] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton, The open quantum materials database (oqmd): assessing the accuracy of dft formation energies, npj Computational Materials 1, 1 (2015).
[4] S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko, and D. Morgan, Aflow:
An automatic framework for high-throughput materials discovery, Computational Materials Science 58, 218 (2012).

[5] P. B. de Castro, K. Terashima, T. D. Yamamoto, Z. Hou, S. Iwasaki, R. Matsumoto, S. Adachi, Y. Saito, P. Song, H. Takeya, and Y. Takano, Machine-learning-guided discovery of the gigantic magnetocaloric effect in hob sub 2 /sub near the hydrogen liquefaction temperature, NPG Asia Materials 12, 35 (2020).

[6] Y. Xiong, Q. T. Campbell, J. Fanghanel, C. K. Badding, H. Wang, N. E. Kirchner-Hall, M. J. Theibault, I. Timrov, J. S. Mondschein, K. Seth, R. Katz, A. M. Villarino, B. Pamuk, M. E. Penrod, M. M. Khan, T. Rivera, N. C. Smith, X. Quintana, P. Orbe, C. J. Fennie, S. Asem-Hiablie, J. L. Young, T. G. Deutsch, M. Cococcioni, V. Gopalan, H. D. Abreuña, R. E. Schaaak, and I. Dabo, Optimizing accuracy and efficacy in data-driven materials discovery for the solar production of hydrogen, Energy & Environmental Science 14, 2335 (2021).

[7] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, Digital twin in industry: State-of-the-art, IEEE Transactions on Industrial informatics 15, 2405 (2019).

[8] Y. Dang, C. Zhu, M. Ikumi, M. Takaishi, W. Yu, W. Huang, X. Liu, K. Kutsukake, S. Harada, M. Tagawa, and T. Ujihara, Adaptive process control for crystal growth using machine learning for high-speed prediction: application to sic solution growth, CrystEngComm 23, 1982 (2021).

[9] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals, and Systems 2, 303 (1989).

[10] N. Mhaskar, H. and A. Micchelli, C., Approximation by superposition of sigmoidal and radial basis functions, Advances in Applied Mathematics 13, 350 (1992).

[11] N. Mhaskar, H, Optimal approximation of continuous functions by very deep relu networks, Neural Computation 8, 164 (1996).

[12] T. Suzuki, Adaptivity of deep reLU network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality, in International Conference on Learning Representations (2019).

[13] M. Imaizumi and K. Fukumizu, Deep neural networks learn non-smooth functions effectively, in Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, Vol. 89, edited by K. Chaudhuri and M. Sugiyama (PMLR, 2019) pp. 869–878.

[14] Anaconda software distribution (2020).
[15] F. Chollet et al., Keras (2015).
[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.

[17] Y. Nesterov, A method of solving a convex programming problem with convergence rate mathcal {O}(1/k^2), in Sov. Math. Dokl, Vol. 27.

[18] T. Tieleman, G. Hinton, et al., Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural networks for machine learning 4, 26 (2012).

[19] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[20] A. Krizhevsky, One weird trick for parallelizing convolutional neural networks, arXiv preprint arXiv:1404.5997 2014 (2014).

[21] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, Accurate, large minibatch sgd: Training imagenet in 1 hour (2017).

[22] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. J. Storkey, Width of minima reached by stochastic gradient descent is influenced by learning rate to batch size ratio, in ICANN (3) (2018) pp. 392–402.

[23] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019).

[24] M. Widenius, D. Axmark, and K. Arno, MySQL reference manual: documentation from the source ("O’Reilly Media, Inc.", 2002).

[25] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, Hyperopt: a python library for model selection and hyperparameter optimization, Computational Science Discovery 8, 014008 (2015).

[26] Streamlit, software available from streamlit.io.
[27] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar, A system for massively parallel hyperparameter tuning, arXiv preprint arXiv:1810.05934 10.48550/ARXIV.1810.05934 (2018).

[28] J. W. Ross and J. Crangle, Magnetization of cubic laves phase compounds of rare earths with cobalt, Phys. Rev. 133, A509 (1964).

[29] H. Wada, S. Tomekawa, and M. Shiga, Magnetocaloric properties of a first-order magnetic transition system erco2, Cryogenics 39, 915 (1999).

[30] K. A. Gschneidner Jr., V. K. Pecharsky, and A. O. Tsokol, Recent developments in magnetocaloric materials, Reports on Progress in Physics 68, 1479 (2005).

[31] X. Tang, H. Sepehri-Amin, N. Terada, A. Martin-Cid, I. Kurniawan, S. Kobayashi, Y. Kotani, H. Takeya, J. Lai, Y. Matsushita, T. Ohkubo, Y. Miura, T. Nakamura, and K. Hono, Magnetic refrigeration material operating at a full temperature range required for hydrogen liquefaction, Nature Communications 13, 1817 (2022).

[32] N. Kawamiya and K. Adachi, Magnetic and m"ossbauer studies of metamagnetic fe3ga4, Journal of the Physical Society of Japan 55, 634 (1986), https://doi.org/10.1143/JPSJ.55.634.

[33] R. Matsumoto, S. Nakano, S. Yamamoto, and Y. Takano, Synthesis and electrical transport measurement of superconducting hydrides using diamond anvil cell with boron-doped diamond electrodes, Japanese Journal of Applied Physics 60, 090902 (2021).

[34] Y. Kim, D. Oh, S. Huh, D. Song, S. Jeong, J. Kwon, M. Kim, D. Kim, H. Ryu, J. Jung, W. Kyung, B. Sohn, S. Lee, J. Hyun, Y. Lee, Y. Kim, and C. Kim, Deep learning-based statistical noise reduction for multidimensional spectral data, Review of Scientific Instruments 92, 073901 (2021).

[35] F. Restrepo, J. Zhao, and U. Chatterjee, Denoising and feature extraction in photoemission spectra with variational auto-encoder neural networks (2022).

[36] A. Damascelli, Z. Hussain, and Z.-X. Shen, Angle-resolved photoemission studies of the cuprate superconductors, Reviews of Modern Physics 75, 473 (2003).

[37] J. Kim, J. K. Lee, and K. M. Lee, Accurate image super-resolution using very deep convolutional networks, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) pp. 1646–1654.

[38] X. Chai, H. Gu, F. Li, H. Duan, X. Hu, and K. Lin, Deep learning for irregularly and regularly missing data reconstruction, Scientific Reports 10, 3302 (2020).
[39] H. Peng, X. Gao, Y. He, Y. Li, Y. Ji, C. Liu, S. A. Ekahana, D. Pei, Z. Liu, Z. Shen, and Y. Chen, Super resolution convolutional neural network for feature extraction in spectroscopic data, Review of Scientific Instruments 91, 033905 (2020), https://doi.org/10.1063/1.5132586.

[40] M. Nagao, A. Miura, S. Demura, K. Deguchi, S. Watauchi, T. Takei, Y. Takano, N. Kumada, and I. Tanaka, Growth and superconducting properties of f-substituted robis2 (r=la, ce, nd) single crystals, Solid State Communications 178, 33 (2014).

[41] M. Kitamura, S. Souma, A. Honma, D. Wakabayashi, H. Tanaka, A. Toyoshima, K. Amemiya, T. Kawakami, K. Sugawara, K. Nakayama, K. Yoshimatsu, H. Kumigashira, T. Sato, and K. Horiba, Development of a versatile micro-focused angle-resolved photoemission spectroscopy system with kirkpatrick–baez mirror optics, Review of Scientific Instruments 93, 033906 (2022).

[42] S. Yamamoto, R. Matsumoto, S. Adachi, and Y. Takano, High-pressure effects on la(o,f)BiS$_2$ single crystal using diamond anvil cell with dual-probe diamond electrodes, Applied Physics Express 14, 043001 (2021).
Supporting Information for
Neural networks for a quick access to a digital twin of scanning physical properties measurements

Kensei Terashima,1,* Pedro Baptista de Castro,1,2 Miren Garbiñe Esparza Echevarria,1,2 Ryo Matsumoto,1 Takafumi D Yamamoto,1 Akiko T Saito,1 Hiroyuki Takeya,1 and Yoshihiko Takano1,2

1National Institute for Materials Science, 1-2-1 Sengen, Tsukuba, Ibaraki, 305-0047 Japan
2University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8577, Japan
Trapezoidal integration is a numerical method to often used to obtain the approximation of the area of interest below the curve of a function $f(x)$ and between the a given set of desired points $a$ and $b$, assuming: $\int_a^b f(x)dx \sim \frac{(b-a)}{2} (f(a) + f(b))$, that corresponds to a summation of linearly-interpolated values of $f(x)$ between starting point $a$ and end point $b$. Sometimes, for some datasets this linear interpolation method can fail. Figure SM1 shows a typical example of such a failure in linear interpolation. Suppose there are data for $y(x_1, x_2)$ taken in $x_2 = a$ and $b$, where peak position of $y$ migrates by changing $x_2$ value, as in Figure SM1(a). If one performs bilinear interpolation as in Figure SM1(b), it does not succeed to reproduce what is desired (as in Figure SM1(a)) but it produces an artificial dip in the obtained interpolated $y(x'_1, x'_2)$. In Section 3.1 of the manuscript, $x_1, x_2, y$ correspond to temperature, magnetic field, and $(\frac{\partial M}{\partial T})$, respectively, namely $y(x_1, x_2) = \frac{\partial M(T,H)}{\partial T}$. This causes extrinsic oscillation in $|\Delta S_M|$ derived by integrating $\left(\frac{\partial M}{\partial T}\right)$ along $H$ direction.

**FIG. SM1.** Schematic graph showing how (bi)linear interpolation fails to grab characteristic change. Top graphs show spectral lineshapes and bottom graphs are corresponding intensity plots. (a) What is expected (b) What (bi)linear interpolation gives.
We also note here that the situation is also valid for isothermal cuts for \( M(T,H) \). In case of isothermal cuts, \( |\Delta S_M| \) can be deduced by \( \Delta S_M(T_i, H) = \sum_i \frac{M_{i+1} - M_i}{T_{i+1} - T_i} \Delta H_i \) (in case of using forward differential), which corresponds to the gray shaded area as depicted in Figure SM2(a). As can be seen in Figure SM2(a) in case of coarse step of \( H \), linear interpolation along \( H \)-direction (corresponds to dotted line in the Figure) creates an extrinsic oscillation of area between two subsequent measured temperatures. Figure SM2(b) shows the simulated \( M(T, H) \) by neural networks model built by Figure SM2(a) train data, where such extrinsic oscillation of area between measured temperatures is suppressed.

![FIG. SM2. Schematic graph showing how linear interpolation along \( H \) affects estimated \( |\Delta S_M(T_i, H)| \) value, in case of isothermal measurement. (a) Coarse magnetic field step (train data) case. Difference in color stands for different measurement temperatures. Gray area corresponds to estimated \( |\Delta S_M(T_i, H)| \). (b) Fine magnetic field step case, simulated by neural networks model.](image)

**OPERATING SYSTEM AND INSTALLATION GUIDE**

The current proposed method have been tested only on Linux and MacOS (Intel) systems. We currently do not support Windows systems, however it might be possible to run using virtual machines, although not recommended. In more details, the following OSes were tested by the authors: Ubuntu LTS 20.04 and 18.04, CentOS 7, and MacOS 11 and 12 (both Intel). A tutorial for how to get ready and perform the neural networks learning and simulation
EXAMPLE OF DURATION TO PERFORM NEURAL NETWORKS LEARNING

Table SM1 shows the total duration it took for each computer to finish 30 trials of learning. The used computers are the following: (i) Intel i9-9880H 2.3 GHz, (ii) Xeon Silver 4116 2.1 GHz, and (iii) Intel i7 10700K 3.8 GHz. The number of parallel workers are set to be 5 for (i) and (iii), while it was 15 for (ii). Since the result would be influenced by random seeds, we tried 5 times for each, and the averaged values are shown in the table. Standard deviation of 5 times of 30 trials are also presented as numbers in parenthesis. The number of train data points are 808, 2737, 767, 29977 for Section 3.1, 3.2, 3.3, 3.4, respectively.

| Data       | Intel i9-9880H 2.3 GHz 8 cores, MacOS11(Intel) | Xeon Silver 4116 2.1 GHz 12x2 cores, CentOS7.6 | Intel i7 10700K 3.8 GHz 8 cores, Ubuntu20.04 |
|------------|-----------------------------------------------|-----------------------------------------------|-----------------------------------------------|
| Section3.1 | 3m 30s (40s)                                 | 3m 53s (29s)                                 | 2m 36s (45s)                                 |
| Section3.2 | 4m 30s (46s)                                 | 3m 17s (26s)                                 | 2m 31s (12s)                                 |
| Section3.3 | 2m 53s (25s)                                 | 3m 8s (30s)                                  | 2m 5s (35s)                                  |
| Section3.4 | 13m 36s (2m 38s)                             | 6m 47s (47s)                                 | 11m 14s (1m 46s)                             |

TABLE SM1. Comparison of ellapsed time required for PCs to finish learning through 30 trials of Bayesian optimization for each dataset shown in the main text. The results are averaged values of 5 times of trials. Numbers in parenthesis stand for standard deviation.

STABILITY OF LEARNED AND SIMULATED RESULTS

Here we show how much the learning curves and the resultant simulated value would differ among each 30 trials. Two examples are shown, one for the data presented in Section 3.1 in the main text (Figure SM3), and another for the data presented in Section 3.3 (Figure SM4), where the latter is more influenced by noise. Here each trial is allowed to keep running until it reaches a maximum epoch number of 500, unless it is stopped by either pruning or earlystopping. After 30 trials of Bayesian optimization (initial 10 trials are random), both the $R^2$ score for training and simulated values show a convergence to some extent.
Figure SM4 shows spectral lineshape of ARPES image plot in Section 3.4 of main text, where gray lines are train data and colored lines are simulated ones by neural networks model. The spectra consist a number of peak structures and model can capture the structures as well as sensitivity of the machine such as relatively high intensity region around the edge.
FIG. SM3. Results of learning and simulations in individual cases of 30 trials for ErCo$_2$ (top) and PdH (bottom) data. For each material, case#1 corresponds to the data in the main text. Left: Learning curves. Colors represent different trials. Center: Simulation by learned models that acquired the best $R^2$ score in each case. Gray lines show train data. Right: Simulation by models.
FIG. SM4. Spectral lineshapes of ARPES image plot, shown as Figure 5(a) and (b) in Section 3.4 of main text. Gray lines correspond to train data, while colored lines correspond to simulated ones by learned model.