Scaler Transfer: A Simple and Data-efficient Simulation-to-Real Transfer Scheme for Materials

Yuta YAHAGI[1,2] yuta-yahagi@nec.com
Kiichi OBUCHI[1,2] kiichi-obuchi@nec.com
Fumihiko Kosaka[2] f.kosaka@aist.go.jp
Kota MATSUI[3] matsui.kota.8k@kyoto-u.ac.jp

[1]NEC Corporation, Minato-ku, Tokyo, Japan, 211-8666
[2]National Institute of Advanced Industrial Science and Technology, Tsukuba City, Japan, 305-8568
[3]Kyoto University, Kyoto City, Japan, 606-8501

Abstract

Data scarcity and domain heterogeneity impede simulation-to-real (sim2real) transfer in materials data. We present a simple, data-efficient recipe that couples a domain transformation with two methods: (i) scaler transfer, which shares standardization parameters fitted on transformed source data to robustly scale scarce target data; and (ii) fine-tuning, which pretrains a predictor on the transformed source and adapts it to the target. On the prediction of electrocatalytic activity with the *Open Catalyst Experiment 2024* datasets, the proposed method consistently surpasses baselines and achieves $R^2 > 0.81$ at the best condition. Critically, the scaler transfer significantly improves the performance of few-shot learning, scoring $R^2 = 0.43$ compared to -0.062 for a baseline. This method is not only easy to implement but also model- and task-agnostic, extending the coverage of sim2real transfer in materials informatics.

1 Introduction

Data scarcity is a pervasive obstacle in materials informatics. Experimental synthesis and measurement are costly and subject to external factors, which limit both the quality and quantity of data. As an alternative, large-scale computational datasets have been increasingly used since material simulations such as density functional theory (DFT) are relatively scalable and easy to automate [1, 2, 3].

Transfer learning from simulation to experiment, referred to as a sim2real transfer, is one of the promising methods, and it has already been applied in the field of materials science [4, 5]. However, there is a gap between them as the DFT draws on microscopic insights, whereas most experiments measure macroscopic quantities, limiting the applicability of transfer learning. It is desired to establish a methodology that realizes a sim2real transfer for materials by bridging this domain heterogeneity.

In this work, we propose a simple and data-efficient sim2real transfer scheme based on **domain transformation with scaler transfer**, which shares a standard scaler from domain-transformed computational data to target experimental data. Standardization is a fundamental preprocessing step in data analysis, but in small-data regimes the scaling parameters may not adequately represent the underlying population distribution. Moreover, naively borrowing data from different domains to increase the sample size can introduce bias into the scaling parameters. In the proposed method, the source data are first mapped onto the target domain while correcting for the domain shift between

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

them. Then, a scaler is constructed based on the source data and shared for the target data; it can standardize target data robustly even with a small data region, such as in few-shot learning.

To evaluate the effectiveness of this method, we conduct an example of catalyst activity prediction on the hydrogen evolution reaction (HER) by using the *Open Catalyst Experiment 2024* (OCx24) dataset that includes both experimental and computational data [6].

The key contributions of this work are as follows:

- **Domain transformation with scaler transfer**: We introduce an efficient knowledge transfer scheme that significantly improves prediction performance, especially in few-shot learning.
- **Widespread positive transfer**: We validate positive transfer across wide range of target data size, underscoring the effectiveness of our approach in a variety of situations.
- **Accurate prediction of HER activity**: The proposed method provides a highly accurate predictive model for HER activity with $R^2 > 0.8$, which outperforms other methods and previous reports.

2 Method

2.1 Problem formulation and notation

Let the source and target dataset be $\mathcal{D}_r \equiv \{(x_r^{(i)}, y_r^{(i)})\}_{i=1}^{N_r} \subset (\mathcal{X}_r \times \mathcal{Y}_r)$, with the feature space $\mathcal{X}^r$, the label space $\mathcal{Y}^r$, the size of dataset N_r where $r = \mathrm{S}$ or T represent the source or target, respectively. Their domains are not required to be identical, namely, both $\mathcal{X}_\mathrm{S} \neq \mathcal{X}_\mathrm{T}$ and $\mathcal{Y}_\mathrm{S} \neq \mathcal{Y}_\mathrm{T}$ are possible. The goal is to predict y_T accurately under small N_T assuming $N_\mathrm{S} \gg N_\mathrm{T}$.

2.2 Proposal: Scaler transfer combined with domain transformation

The proposed method consists of two steps: domain transformation [7] and knowledge transfer.

The domain transformation maps the source data onto the target domain guided by chemistry knowledge, for instance, a linear relationship between the source label and the target label. This process approximately reduces the domain heterogeneity to a co-variate shift, which can be solved through transfer learning.

Let $\mathcal{D}_{\mathrm{S}'} \subset \mathcal{X}_\mathrm{T} \times \mathcal{Y}_\mathrm{T}$ be the transformed source data, the second step performs transfer learning from $\mathcal{D}_{\mathrm{S}'}$ to $\mathcal{D}_\mathrm{T}$ with two transfer schemes: **scaler transfer** and **fine-tuning**.

- **Scaler transfer (ST):** Fit a standard scaler with $\mathcal{D}_{\mathrm{S}'}$, then reuse it for $\mathcal{D}_\mathrm{T}$.
- **Fine-tuning (FT):** Pretrain a model with $\mathcal{D}_{\mathrm{S}'}$, then retrain the same instance with $\mathcal{D}_\mathrm{T}$.

3 Demonstration: HER catalyst activity prediction

3.1 Dataset and domain transformation

In this study, we employ the OCx24 dataset [6] for both source and target. The source (computational) dataset is represented as $\mathcal{D}_\mathrm{S} \equiv \{(x_\mathrm{S}^{(j)}, y_\mathrm{S}^{(j)})\}_{i=1}^{N_{src}}$, where x_S specifies a surface structure (base material, facet, adsorption site) and $y_\mathrm{S} = \left(y_{\mathrm{S,H}}, y_{\mathrm{S,OH}}\right)$ are adsorption energies of H and OH derived from DFT. The size of the source dataset is approximately $N_\mathrm{S} \sim 700,000$.

The target (experimental) dataset is represented as $\mathcal{D}_\mathrm{T} \equiv \{(x_\mathrm{T}^{(j)}, y_\mathrm{T}^{(j)})\}_{i=1}^{N_\mathrm{T}}$, where x_T is chemical composition and y_T is measured potential relevant to the activity of HER. The size of the target dataset is $N_\mathrm{T} = 272$. Further details are explained in Appendix A.

To address a domain heterogeneity between the source and target, we transform $\mathcal{D}_\mathrm{S}$ to $\mathcal{D}_{\mathrm{S}'} \subset (\mathcal{X}_\mathrm{T} \times \mathcal{Y}_\mathrm{T})$ with the following procedure [7]:

(1) Averaging $\left(\mathcal{X}_\mathrm{S} \times \mathcal{Y}_\mathrm{S} \to \mathcal{X}_\mathrm{T} \times \mathcal{Y}_\mathrm{S}\right)$**.** For a given composition c, we aggregate corresponding source entries with $\mathcal{I}(c) = \{\, i : \ \mathrm{composition}(x_\mathrm{S}^{(i)}) = c \,\}$, and obtain averaged labels $\tilde{\mathbf{g}}(c) =$

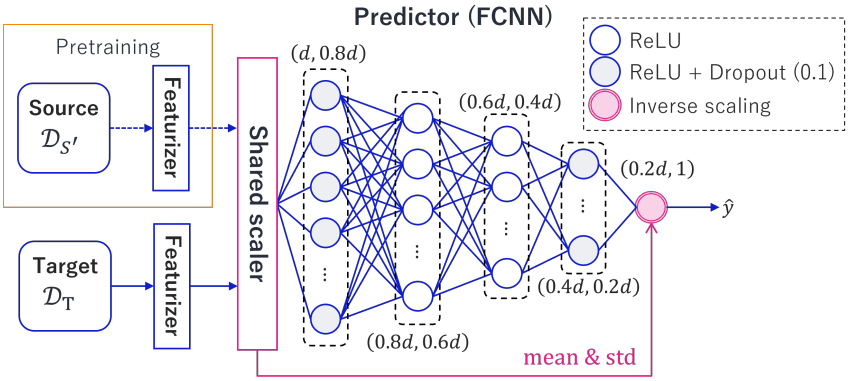

Figure 1: Architecture of the prediction model. d is dimension of feature vector.

$(\bar{g}_{\mathrm{H}}, \bar{g}_{\mathrm{OH}})$ by

$$\bar{g}_a(c) = \frac{1}{|\mathcal{I}(c)|} \sum_{i \in \mathcal{I}(c)} y_{\mathrm{S},a}^{(i)}, \quad a \in \{\mathrm{H}, \mathrm{OH}\}. \tag{1}$$

(2) Function estimation $\left(\mathcal{Y}_{\mathrm{S}} \to \mathcal{Y}_{\mathrm{T}}\right)$**.** We then map $\tilde{\mathbf{g}} = (\bar{g}_{\mathrm{H}}, \bar{g}_{\mathrm{OH}})$ to y_{T} using a linear regressor,

$$F_\theta(\tilde{\mathbf{g}}) = \theta_0 + \theta_1 \, \bar{g}_{\mathrm{H}} + \theta_2 \, \bar{g}_{\mathrm{OH}}, \tag{2}$$

obtained by minimizing the mean-squared error loss, $l_{\mathrm{MSE}}\left(\tilde{\mathbf{g}}(x_{\mathrm{T}}), y_{\mathrm{T}}\right)$.

(3) Domain transformation. Using F_θ, we transform all entries of $\mathcal{D}_{\mathrm{S}}$ to $\mathcal{D}_{\mathrm{S}'} = \{x_{\mathrm{S}'}^{(i)}, y_{\mathrm{S}'}^{(i)}\}_i^{N_{\mathrm{S}'}} \equiv \left\{c^{(i)}, F_\theta(\tilde{\mathbf{g}}(c^{(i)}))\right\}_{i=1}^{N_{\mathrm{S}'}}$. Here, the size of the transformed dataset is $N_{\mathrm{S}'} = 12,463$. By definition, $N_{\mathrm{S}'}$ is equal to the number of varieties of composition contained in $\mathcal{D}_{\mathrm{S}}$. During this procedure, 43 target data are used for function estimation. To avoid a data leak, these data will be excluded from the test set. Further detail is explained in Appendix B.

3.2 Predictive model and training pipeline

Overall architecture of the predictive model is shown in Fig. 1. To optimize the model, we first pretrain the model with $\mathcal{D}_{\mathrm{S}'}$ and retrain it with $\mathcal{D}_{\mathrm{T}}$.

The featurizer encodes a composition string into chemical descriptors. In this work, we use 186 features among XenonPy composition descriptors [8] (See Appendix C).

Both features and labels are standardized with a scaler. The scale parameter is estimated from $\mathcal{D}_{\mathrm{S}}$ in the pretraining step and shared in the target training step.

As a predictor, we employ a fully connected network (FCNN), a common prediction head in neural net architectures, implemented in PyTorch framework [9]. It consist with five layers and the number of units in each layer gradually decreases, starting from the input dimension 186 in the first layer, followed by 148, 111, 74, and 37 in the middle layers, with the final output layer having one unit. All units are activated by ReLU (rectified linear unit). During training, dropout affects the first and fourth layers at a rate of 0.1. The final unit is inversely scaled after training is completed. Detailed (pre-)training conditions are shown in Appendix D.

3.3 Evaluation protocol

The effects of sim2real transfer are measured by the determination coefficient R^2 as a function of the size of the target training dataset. Here, 20% of $\mathcal{D}_{\mathrm{T}}$ were used to evaluate R^2 and the remaining 80% were used to train the predictive model. For a given data size, we adopt the average value of R^2 obtained from the randomly selected training set over a certain number of trials.

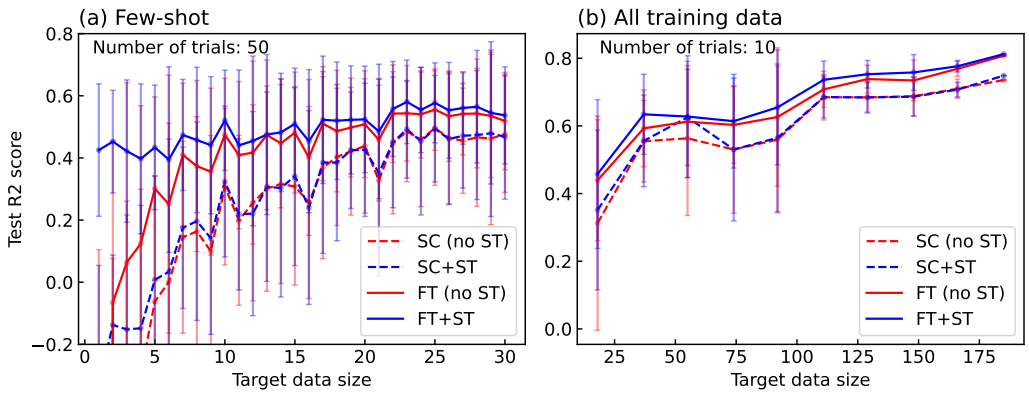

Figure 2: Test R^2 scores as functions of the size of target training data for each training method of fine-tuning (FT) or learning from scratch (SC), with or without the scale transfer (ST).

Table 1: Summary of R^2 score for HER activity prediction.

Method	Best score	Few-shot (5 data) score
Linear model [6]	0.63	-
FCNN	0.71	-0.062
FCNN+FT+ST (This work)	**0.81**	**0.43**

4 Results

We quantify the contribution of each transfer scheme in Sec. 2.2 by comparing fine-tuning (FT) and learning from scratch (SC) with or without the scaler transfer (ST) for the target label, as represented in Fig. 2. Note that the feature scaler is reused even in the case of no ST, because it can be easily obtained using unlabeled data in practice.

In the small data regime (Fig. 2(a)), the proposed method (FT+ST) showed a significant improvement in prediction performance. The model trained by FT performs approximately 0.1-0.2 higher R^2 than that trained by SC, and the effect becomes more pronounced as the datasize decreases. It is noteworthy that the ST dramatically enhanced the effect of FT in few-shot learning (less than 10 data).

Increasing the size of the target data (Fig. 2(b)), all methods improve as data grow, but FT+ST remains best throughout and achieves $R^2 > 0.8$ by using all training data. Although the effect of ST diminishes compared to the few-shot regime, there is still a slight improvement in performance.

Table 1 summarizes R^2 for each method and clearly indicates that the proposed method (FT+ST) outperforms other methods, including the previous work [6].

5 Summary

We present a simple and data-efficient sim2real transfer scheme: **scale transfer (ST)** is a method that reuses a scaler fitted on the source data to standardize target data robustly even in few-shot learning; **fine-tuning (FT)** is a method that pretrains with the source data and then retrains the same instance with target data, providing a nearly optimized model while correcting the residual domain gap. To evaluate, we predict the hydrogen evolution reaction (HER) catalyst activity in the *Open Catalyst Expriment 2024* dataset [6] using this method combined with a domain transformation technique [7].

Across target data sizes, **FT** consistently outperforms training from scratch (SC), and **ST** further improves prediction performance—most markedly in the few-shot regime (less than 10 data). The combined **FT+ST** configuration yields the strongest and most stable gains: in the small-data regime it performs 0.1-0.2 higher R^2 than that of SC, and with all training data it attains $R^2 > 0.8$.

While the present study demonstrated it on the HER catalysts, the proposed method can be similarly applied to other materials and properties. In addition, standardizing labels allows using common learning conditions (such as learning rate, regularization parameter, etc.) for learning different physical properties in different units. Furthermore, this method is model-agnostic and not limited to neural networks. This versatility is a clear advantage.

A limitation of our approach is the requirement for compositional overlap between simulations and experiments to enact the domain transformation. This can be mitigated by designing experiments including compositions already represented in large computational datasets, or by augmenting simulation datasets to cover compositions appearing in measurements. Moreover, in practical cases, it is common that laboratory equipments are calibrated with some standard materials, providing some overlap between calculations and experiments on such materials. By leveraging a small number of overlaps, it is possible to transfer the scale advantages of abundant computational data to real-world tasks, thereby reducing the number of experiments that consume budget, time, and human resources.

Acknowledgement

This work was supported by Grants-in-Aid from the Japan Society for the Promotion of Science (JSPS) for Scientific Research (KAKENHI grant nos. JP23K28146, JP24K20836, JP25K03086 to K.M.).

References

[1] Stefano Curtarolo, Gus L W Hart, Marco Buongiorno Nardelli, Natalio Mingo, Stefano Sanvito, and Ohad Levy. The high-throughput highway to computational materials design. Nat. Mater., 12(3):191–201, March 2013.

[2] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C Lawrence Zitnick, and Zachary Ulissi. Open catalyst 2020 (OC20) dataset and community challenges. ACS Catal., 11(10):6059–6072, May 2021.

[3] Richard Tran, Janice Lan, Muhammed Shuaibi, Brandon M Wood, Siddharth Goyal, Abhishek Das, Javier Heras-Domingo, Adeesh Kolluru, Ammar Rizvi, Nima Shoghi, Anuroop Sriram, Félix Therrien, Jehad Abed, Oleksandr Voznyy, Edward H Sargent, Zachary Ulissi, and C Lawrence Zitnick. The open catalyst 2022 (OC22) dataset and challenges for oxide electrocatalysts. ACS Catal., 13(5):3066–3084, March 2023.

[4] Dipendra Jha, Kamal Choudhary, Francesca Tavazza, Wei-Keng Liao, Alok Choudhary, Carelyn Campbell, and Ankit Agrawal. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. Nat. Commun., 10(1):5316, November 2019.

[5] Stephen Wu, Yukiko Kondo, Masa-Aki Kakimoto, Bin Yang, Hironao Yamada, Isao Kuwajima, Guillaume Lambard, Kenta Hongo, Yibin Xu, Junichiro Shiomi, Christoph Schick, Junko Morikawa, and Ryo Yoshida. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. Npj Comput. Mater., 5(1):1–11, June 2019.

[6] Jehad Abed, Jiheon Kim, Muhammed Shuaibi, Brook Wander, Boris Duijf, Suhas Mahesh, Hyeonseok Lee, Vahe Gharakhanyan, Sjoerd Hoogland, Erdem Irtem, Janice Lan, Niels Schouten, Anagha Usha Vijayakumar, Jason Hattrick-Simpers, John R Kitchin, Zachary W Ulissi, Aaike van Vugt, Edward H Sargent, David Sinton, and C Lawrence Zitnick. Open catalyst experiments 2024 (OCx24): Bridging experiments and computational models. arXiv [cond-mat.mtrl-sci], November 2024.

[7] Yuta Yahagi, Kiichi Obuchi, Fumihiko Kosaka, and Kota Matsui. Transfer learning from first-principles calculations to experiments with chemistry-informed domain transformation. Mach. Learn. Sci. Technol., 6(2):025026, June 2025.

[8] Chang Liu, Erina Fujita, Yukari Katsura, Yuki Inada, Asuka Ishikawa, Ryuji Tamura, Kaoru Kimura, and Ryo Yoshida. Machine learning to predict quasicrystals from chemical compositions. Adv. Mater., 33(36):e2102507, September 2021.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. arXiv [cs.LG], December 2019.

A Dataset description

Throughout our demonstration, we refer OCx24 dataset for both the source and the target dataset. All data can be found in their repository (https://github.com/facebookresearch/fairchem/). Details of the computational and experimental pipeline can be found in the original paper [6].

As the source dataset, we employ the computational dataset in OCx24 generated by a high-throughput computation workflow. This dataset includes both values obtained from DFT calculations and inferences made by machine learning models. In this study, we use only the former for the sake of accuracy. It contains approximately 700,000 entries, and each entry represents a surface structure (x_{S}) and adsorption energies of H and OH, $(y_{\mathrm{S,H}}, y_{\mathrm{S,OH}})$.

As the target dataset, we employ the experimental dataset in OCx24, specifically the data in ExpDataDump_YYMMDD_clean.csv. This dataset is prepared with a high-throughput pipeline of synthesis, characterization, and measurement.

The compositional overlaps among these datasets can be found in HER_40_70_matched.csv.

B Details of domain transformation

Using the compositional overlaps among $\mathcal{D}_{\mathrm{S}}$ and $\mathcal{D}_{\mathrm{T}}$, we attempt domain transformations with the following conditions:

(1) Averaging (two variants).

$$\textbf{Mean (arithmetic average):} \quad \bar{g}_a = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} y_{\mathrm{S},a}^{(i)}, \quad a \in \{H, \mathrm{OH}\}; \tag{3}$$

$$\textbf{Boltzmann average:} \quad \bar{g}_a = \sum_{i \in \mathcal{I}} \frac{\exp(-\beta y_{\mathrm{S},a}^{(i)})}{\sum_{k \in \mathcal{I}} \exp(-\beta y_{\mathrm{S},a}^{(k)})} \, y_{\mathrm{S},a}^{(i)}, \tag{4}$$

with $\beta = 38.68\ \mathrm{eV}^{-1}$ at room temperature (300 K).

(2) Function estimation (two variants).

$$\textbf{Linear model:} \quad F_\theta(\tilde{\mathbf{g}}) = \theta_0 + \theta_1 \, \bar{g}_{\mathrm{H}} + \theta_2 \, \bar{g}_{\mathrm{OH}}, \tag{5}$$

$$\textbf{Piecewise LASSO (PWLASSO):} \quad F_\theta(\tilde{\mathbf{g}}) = \mathsf{PWLASSO}(g_{\mathrm{H}}, \, \bar{g}_{\mathrm{OH}}). \tag{6}$$

Regression results are shown in Figs. 3-6.

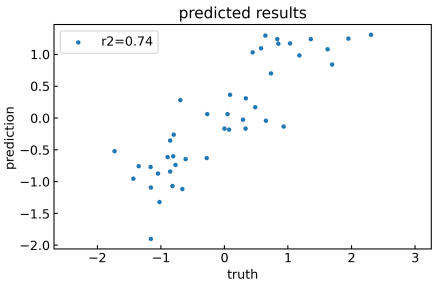

Figure 3: Result of **Mean-Linear**.

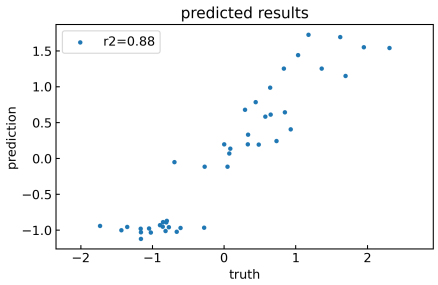

Figure 4: Result of **Mean-PWLASSO**.

According to the results, **Mean-PWLASSO** (Fig. 4) scores the best R^2; however, it shows an unphysical concentration of data. Thus, we finally adopt **Mean-Linear** (Fig. 3) as the next best.

Figs. 7 represent the label distribution of original source data and 8 represent the transformed one compared with that of the target data. It is clear that the domain conversion has made it possible to compare values that were different not only in terms of numerical values but also in terms of units.

C Compositional featurizer

The XenonPy compositional featurizer is a tool within the XenonPy library that converts raw chemical composition data into informative numerical descriptors suitable for machine learning [8].

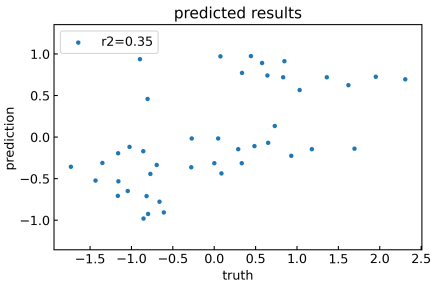

Figure 5: Result of **Boltzmann-Linear**.

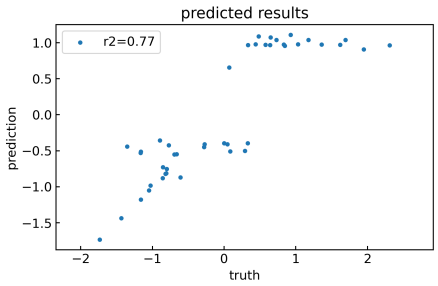

Figure 6: Result of **Boltzmann-PWLASSO**.

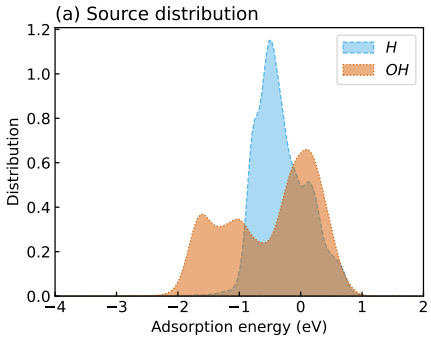

Figure 7: Label distribution of source data.

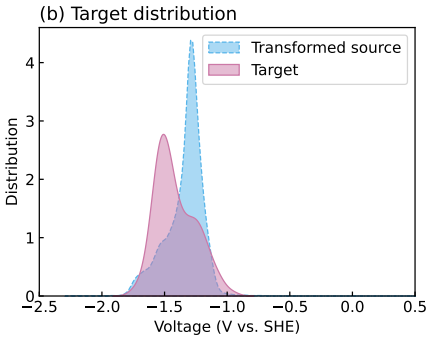

Figure 8: Label distribution of terget data compared with the transformed source.

It can calculate 290 compositional features for a given chemical composition. For more details, see the code documentation (https://xenonpy.readthedocs.io/en/latest/features.html).

The compositional descriptors are obtained from the five calculations, weighted-sum, weighted-average, weighted-variance, max-pooling, and min-pooling. In this work, we omit the weighted-sum calculation as it is identical to the weighted-average for a periodic system.Moreover, inspecting the features in source data, we omit low-variance features (standard deviation < 0.01) and highly correlated features (threshold $= 0.95$), resulting in 186 features, eventually.

D Training details

All essential parameters from our experiments are summarized in Table 2.

Figs 9 and 10 show training curves on pretraining and target training, respectively.

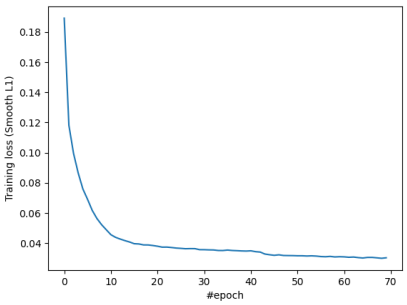

Figure 9: Learning curve on pretraining.

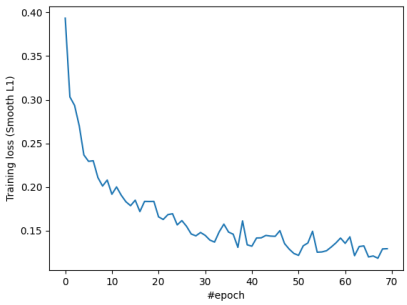

Figure 10: Learning curve on target training

Table 2: List of model parameters and training conditions of our demonstration.

Parameter	Value
Implementation	PyTorch [9]
Input dimension	186
Output dimension	1
Training batch size	32
Optimizer	AdamW
Initial learning rate	0.001
Weight decay	0.0005
Loss function	Smooth L1 loss
Scheduler	ReduceLROnPlateau
Random seed	42
Epochs (source pretrain)	70
Epochs (target training)	70