Injecting Domain Knowledge from Empirical Interatomic Potentials to Neural Networks for Predicting Material Properties

Zeren Shui
University of Minnesota
shuix007@umn.edu

Daniel S. Karls*
University of Minnesota
karl0100@umn.edu

Mingjian Wen*
University of Houston
mjwen@uh.edu

Ilia A. Nikiforov
University of Minnesota
nikif002@umn.edu

Ellad B. Tadmor
University of Minnesota
tadmor@umn.edu

George Karypis
University of Minnesota / AWS
karypis@umn.edu

Abstract

For decades, atomistic modeling has played a crucial role in predicting the behavior of materials in numerous fields ranging from nanotechnology to drug discovery. The most accurate methods in this domain are rooted in first-principles quantum mechanical calculations such as density functional theory (DFT). Because these methods have remained computationally prohibitive, practitioners have traditionally focused on defining physically motivated closed-form expressions known as empirical interatomic potentials (EIPs) that approximately model the interactions between atoms in materials. In recent years, neural network (NN)-based potentials trained on quantum mechanical (DFT-labeled) data have emerged as a more accurate alternative to conventional EIPs. However, the generalizability of these models relies heavily on the amount of labeled training data, which is often still insufficient to generate models suitable for general-purpose applications. In this paper, we propose two generic strategies that take advantage of unlabeled training instances to inject domain knowledge from conventional EIPs to NNs in order to increase their generalizability. The first strategy, based on weakly supervised learning, trains an auxiliary classifier on EIPs and selects the best-performing EIP to generate energies to supplement the ground-truth DFT energies in training the NN. The second strategy, based on transfer learning, first pretrains the NN on a large set of easily obtainable EIP energies, and then fine-tunes it on ground-truth DFT energies. Experimental results on three benchmark datasets demonstrate that the first strategy improves baseline NN performance by 5% to 51% while the second improves baseline performance by up to 55%. Combining them further boosts performance.

1 Introduction

Predictive modeling of materials is a field with manifold applications that has been the subject of many cross-disciplinary studies. While modeling is conducted at different length scales, all material behavior ultimately has its origins at the nano scale, where the interactions between individual atoms must be understood. The most accurate methods at this scale are based on quantum mechanics theory, requiring explicit consideration of the electronic degrees of freedom described by the Schrödinger equation. However, these methods are presently limited to systems containing at most several

*Equal contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

thousand atoms, precluding their use in investigation of important microstructural phenomena such as crack propagation. To overcome this limitation, practitioners have long relied upon heuristic models known as *empirical interatomic potentials* (EIPs), which consist of physically motivated analytical functional forms that strive to model the complex electronic interactions between atoms using only the nuclear coordinates of the atoms and their elemental species as input [40]. In the past several years, there has been a surge of interest in the development of machine learning EIPs, particularly those based on neural networks (NNs), as a more accurate alternative to traditional EIPs [4]. In contrast to traditional EIPs, NN potentials contain little inductive bias and, accordingly, require large volumes of training data labeled using first-principles quantum mechanical methods. The first-principles method most commonly used is density functional theory (DFT), which scales as $\mathcal{O}(n_{\mathrm{e}}^3)$ with the number of valence electrons n_{e} in a given configuration of atoms [26]. As a result of this high computational cost, it is difficult to acquire a sufficient number of labeled training instances to create an NN potential that performs accurately over a wide range of applications.

One potential solution to this problem is to seek additional supervision signals. Traditional EIPs are attractive sources of such additional supervision for two reasons. First, their functional forms incorporate prior physical information that allows them to correlate with DFT, and, in regions of relevance to common applications, are often quite accurate. That is, they contain domain knowledge that could benefit the training of NN potentials. Second, EIPs scale linearly with the number of atoms and are thus orders of magnitude faster than DFT, permitting the labeling of massive datasets. Despite the advantages, no research has focused on using EIP supervision signals in training NN potentials.

In this paper, we leverage physically motivated EIPs and unlabeled configurations to tackle the label scarcity challenge for training NN potentials.We propose two generic strategies, weakly supervised learning and transfer learning, for exploiting this additional source of information. In the first strategy, we expand the DFT-labeled training set with unlabeled configurations and their EIP energies. To achieve this goal, we train an auxiliary classifier on the original DFT-labeled training configurations that predicts which one of a selected set of EIPs is likely to produce the most accurate estimate of the DFT energy for each of a large set of unlabeled configurations. The unlabeled configurations are then labeled by their corresponding predicted best-performing EIPs and appended to the training set. We train NN potentials on the expanded training set by optimizing a robust regression loss to mitigate the influence of noise and outliers introduced by the EIP energies. In the second strategy, we adopt a transfer learning approach by way of multi-task pretraining. We first pretrain the representation module of an NN potential to reproduce the energies predicted by the EIPs. During the subsequent fine-tuning stage, the representation module of the NN is paired with a prediction head and trained on the DFT-labeled configurations. These two strategies can be flexibly used and coupled to train any NN potential.

The contributions of this work are three-fold: 1) We demonstrate that EIPs are capable of providing high quality supervision signals for training NN potentials, which opens a new direction for future development of NN potentials; 2) We propose two effective and generic strategies that take advantage of EIPs and unlabeled configurations to tackle the label scarcity challenge for training NN potentials; 3) We conduct comprehensive experiments on three benchmark datasets and four representative NN potentials that cover most of the NN potential forms currently in use. Experimental results show that the proposed strategies successfully inject domain knowledge from EIPs to NN potentials and improve the performance of the NN potentials by up to 55%.

2 Preliminaries

2.1 Atomic Configurations

The fundamental input in atomistic modeling is an *atomic configuration*. An atomic configuration is a spatial arrangement of atoms $C = \{(Z_i, \mathbf{r}_i)\}_{i=1}^N$ where Z_i and $\mathbf{r}_i$ are the atomic number and the three-dimensional Euclidean coordinates of atom i, respectively. In the simplest scenario, an atomic configuration corresponds to an isolated cluster of atoms that comprise a molecule. However, it may more generally describe a bulk system such as a crystal, which contains an infinite number of atoms distributed over space. These systems are modeled using a small collection of atoms in a finite simulation cell that is effectively repeated across all of space with the aid of periodic boundary conditions (PBCs) [40] (see Appendix A.1).

2.2 Physics-based Potentials

Mathematically, an EIP is a function, $E = \mathcal{V}(C; \theta)$, that takes an atomic configuration C as input and returns its total potential energy E; here, θ denotes a set of fitting parameters to be determined. The functional form $\mathcal{V}$ of a physics-based EIP is made up of carefully designed analytic expressions that strive to capture the underlying physics in the material it models [50, 51]. Because the functional form itself is intended to capture most of the relevant physics, such models need relatively few parameters (typically on the order of ten) and are usually fitted to a set of material properties deemed most relevant to real-world applications. It is expected that physics-based EIPs will approximate the first-principles energy surface well in the vicinity of atomic configurations corresponding to the material properties to which they were fit. However, their generalizability can be inconsistent, as shown in Fig. 1.

2.3 Machine Learning Potentials

In contrast to physics-based EIPs, machine learning EIPs employ general-purpose regression algorithms as the functional form $\mathcal{V}$ that do not encode any knowledge of the material it models. Therefore, machine learning EIPs are almost exclusively trained on large sets of DFT data so as to include as much physical knowledge as possible.

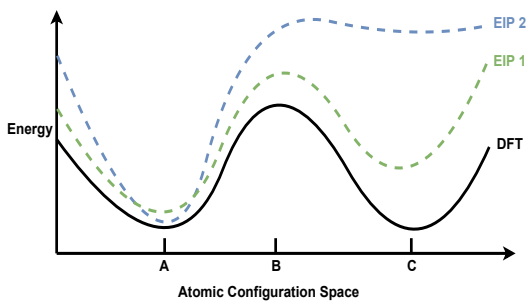


Figure 1: Schematic illustration of the energy landscape defined over atomic configuration space by two physics-based EIPs and by DFT. Both EIPs are fitted to reproduce properties of the DFT energy landscape near atomic configuration A. Away from configuration A, the relative accuracy of the EIPs compared to DFT varies: at point B, EIP 1 is a fair approximation while EIP 2 is less accurate; at point C, neither EIP is accurate.

One important class of machine learning EIPs are neural network (NN)-based potentials. In this paper, we define an NN potential $\mathcal{M} : \mathcal{C} \mapsto \mathcal{E}$ as

$$\mathcal{M}(C) = f_{\text{pred}} \circ f_{\text{rep}}(C),$$

where $f_{\text{rep}} : \mathcal{C} \mapsto \mathcal{R}^{n \times d}$ is the representation learning module that maps each of the n atoms of C to a feature vector of length d based on its local environment, and $f_{\text{pred}} : \mathcal{R}^{n \times d} \mapsto \mathcal{E}$ is the prediction module that maps the feature vectors of the atoms to the total potential energy.

A concrete instance of this type of model are those that use a graph neural network (GNN) [28, 18, 46, 37, 24] as the representation learning module and a multilayer perceptron (MLP) as the prediction module. Before being passed to the representation learning module f_{rep}, an atomic configuration C is first converted to a graph $\mathcal{G} = (V, E)$ with nodes V and edges E. One node is defined for each atom in the simulation cell, as well as for additional padding atoms representing PBCs if present. An edge is created between any two nodes with a distance smaller than a prescribed cutoff radius. Next, for each atom i, its atomic number Z_i is one-hot encoded into an initial feature vector $h_i^{(0)}$ of the node corresponding to the atom. The GNN representation learning module f_{rep} then updates the feature vectors using a message passing paradigm [17], i.e., the nodes iteratively aggregate information from their neighbors. Formally, the feature vector of node i at the $l+1$-th layer $\mathbf{h}_i^{(l+1)}$ is updated as a function of the feature vectors of its neighbors $\mathcal{N}(i)$ and itself at the previous layer,

$$\mathbf{m}_i^{(l)} = \text{Agg}^{(l)}\left(\left\{f^{(l)}\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij}^{(l)}\right) \mid j \in \mathcal{N}(i)\right\}\right), \tag{1}$$

$$\mathbf{h}_i^{(l+1)} = \text{Update}^{(l)}\left(\mathbf{h}_i^{(l)}, \mathbf{m}_i^{(l)}\right), \tag{2}$$

where $f^{(l)}$ and $\text{Update}^{(l)}$ are learnable functions, $\text{Agg}^{(l)}$ is a permutation-invariant function that operates on sets of feature vectors, and $\mathbf{e}_{ij}^{(l)}$ denotes the feature vector associated with the edge connecting node i and node j at the l-th layer. Finally, the prediction module f_{pred} maps the feature vector of each atom at the last layer $\mathbf{h}_i$ to a corresponding energy contribution and sums them to arrive at the total energy of the configuration, i.e., $E = \sum_i \text{MLP}(\mathbf{h}_i)$.

3 Related Works

3.1 Neural Network Potentials

The first modern NN potential was proposed by Behler and Parrinello [3]. In their formulation, an atomic descriptor (i.e., basis functions that transform an atomic configuration into a fixed-length fingerprint vector) based on the bond lengths and bond angles is passed to an MLP. On top of this formulation, a Monte Carlo dropout technique can be applied to the MLP to equip the potential with the ability to quantify its predictive uncertainty [49]. The DeePMD method is a similar approach to that of Behler and Parrinello, except that a novel atomic environment descriptor is used [55]. More recently, researchers have developed GNN potentials based on a message passing paradigm [36, 53]. Another class of GNN potentials such as NequIP [2] and GemNet [16] pass equivariant messages rather than invariant ones based on the formulation of tensor field networks [43] and achieve state-of-the-art performance. All of these models are trained with supervised learning, without exploring the possibilities of leveraging weakly supervised learning or transfer learning to take advantage of unlabeled data.

3.2 Weakly Supervised Learning

Weakly supervised learning refers to techniques that attempt to train machine learning models from incomplete (only a portion of training instances are labeled) or inaccurate (noisy labels) supervision signals [57]. Solutions for incomplete supervision usually fall into the category of semi-supervised learning, which assumes that nearby instances have similar labels [29, 39, 35]. For inaccurate supervision, a model either learns directly from noisy labels with noise-robust algorithms [14, 56, 15] or resorts to a small portion of clean labeled data to reduce the noise [45, 52].

3.3 Transfer Learning

Transfer learning [58] refers to a machine learning paradigm that transfers knowledge a model learns from one or more relevant tasks to benefit a target task. Transfer learning has enjoyed great success, especially in the low-data regime, as demonstrated by the rise of pretrained neural networks. This recent trend began with natural language processing when BERT [10] and successive large pretrained language models [31, 7] were released and quickly gained popularity in other domains such as computer vision [19, 9, 11] and graph learning [54, 22, 33]. These methods pretrain large neural networks on self-supervised tasks in order to encode common contextual knowledge in the structured input. Another approach imparts domain-specific knowledge by pretraining models on tasks related to the target task but for which abundant labeled data is available [21, 32]. The pretrained model is then fine-tuned on the limited training data of the target task.

4 Methods

4.1 Problem Definition

Let $\mathcal{P}$ be a set of physics-based EIPs, $\mathcal{C}$ be the space of all possible atomic configurations, $\mathcal{C}_{\text{DFT}} = \{C_i, \{E_i^p\}_{p\in\mathcal{P}}, E_i^{\text{DFT}}\}_{i=1}^m$ be a set of m configurations with corresponding DFT and EIP energies, and $\mathcal{C}_{\text{EIP}} = \{C_i, \{E_i^p\}_{p\in\mathcal{P}}\}_{i=m+1}^{m+n}$ be a set of n configurations with only physics-based EIP energies. Here, E_i^{DFT} and E_i^p denote the energies predicted for configuration C_i by DFT and the p-th physics-based EIP, respectively. In practice, physics-based EIPs are much less expensive than DFT, and so the size of $\mathcal{C}_{\text{EIP}}$ is much larger than $\mathcal{C}_{\text{DFT}}$, i.e., $n \gg m$. Our goal is to use this data from physics-based EIPs to train an NN-based EIP $\mathcal{M}$ to closely approximate the DFT energy surface over the space of all atomic configurations, i.e.,

$$\mathcal{M}(C) \approx E^{\text{DFT}}, \forall C \in \mathcal{C}.$$

4.2 Label Augmentation

Because physics-based EIPs are developed as approximations to DFT, it is intuitive to use their predictions as surrogate labels for configurations without DFT energies in order to resolve the label scarcity issue of training neural networks. However, there are two fundamental challenges. First, as

discussed in Sec. 2.2, while physics-based EIPs are designed to be accurate in specific regions of the configuration space and generalize better than those based on machine learning, they may still be inaccurate in other regions. Given an arbitrary configuration for which no DFT energy is available, it is unknown which EIP from a given set will yield the most accurate prediction and how large its error will be. Second, surrogating DFT energies with EIP energies inevitably introduces noise, and potentially outliers, into the training set that may have pathological effects.

EIP prediction using an auxiliary classification model. In order to augment training sets with unlabeled configurations and their corresponding EIP-approximated energies, we use an auxiliary classification model to predict the best-performing EIP for a given configuration. For a configuration C, the classification model predicts a discrete distribution over the EIP set $\mathcal{P}' = \mathcal{P} \bigcup \{p_\emptyset\}$ that indicates their probability of being the most accurate EIP for C, i.e., $\mathbb{P}(\mathcal{P}' \mid C)$. We introduce a dummy EIP, $p_\emptyset$, to represent the case where none of the physics-based EIPs in $\mathcal{P}$ is predicted to approximate DFT to an accuracy level c, i.e., $\frac{1}{N_i}||E_i^p - E_i^{\text{DFT}}||_1 > c, \forall p \in \mathcal{P}$, where N_i is the number of atoms in configuration i; throughout this work, c is set to 0.1. By excluding configurations that are labeled with the dummy class from the training set, c acts as a confidence threshold to control the noise and outliers introduced by using the surrogate EIP energies.

The classification model consists of a representation learning module that converts an atomic configuration to fixed-length feature vectors (one for each atom in the configuration), a permutation-invariant readout function that aggregates them to form a feature vector describing the entire configuration, and a prediction module that maps the configuration representation to a set of probabilities. We train the classification model on $\mathcal{C}_{\text{DFT}}$ by optimizing a cross-entropy loss and apply it to $\mathcal{C}_{\text{EIP}}$. Configurations with a predicted EIP other than $p_\emptyset$ are assigned the corresponding EIP energy and are merged with the configurations that have DFT energies to arrive at the final training set. We denote the set of EIP-labeled configurations as $\hat{\mathcal{C}}_{\text{EIP}} = \{C_i, \{E_i^p\}_{p\in\mathcal{P}}, E_i^{\hat{p}_i}\}_{i=m+1}^{m+s}$ where $1 \leq s \leq n$ is the number of selected configurations not labelled by DFT, and $\hat{p}_i$ and $E_i^{\hat{p}_i}$ are the predicted best-performing EIP and its prediction on C_i.

Regression with robust loss functions. We train the NN potential using configurations with ground-truth DFT energies and configurations with EIP energies selected by the classification model. In regression problems, models are usually trained by optimizing the mean square error (MSE) loss. The MSE loss is sensitive to outliers, as the magnitude of its gradient is linearly proportional to the difference between the predicted value and the ground truth value. To lessen the impact of outliers, we optimize the MSE loss on DFT-labeled configurations, while on EIP labeled-configurations, we optimize the Tukey biweight (bisquare) loss [6, 13, 5], i.e.,

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m}(E_i^{\text{DFT}} - \hat{E}_i)^2 + \frac{\alpha}{s}\sum_{i=m+1}^{m+s} l_{\text{Tukey}}(E_i^{\hat{p}_i} - \hat{E}_i) \tag{3}$$

where $\hat{E}_i$ is the model prediction of the energy for configuration C_i, and $\alpha > 0$ is a hyper-parameter that controls the contribution of EIP-labeled configurations to the loss and its gradient.

The Tukey biweight loss falls under the M-estimation method [23] and is intended to screen outliers using robust statistics of the regression residuals such as median absolute residuals (MAR) and suppress their influence on the gradient. The Tukey loss function is computed as

$$l_{\text{Tukey}}(r_i) = \begin{cases} \frac{k^2}{6}\left[1 - \left(1 - \left(\frac{r_i}{k}\right)^2\right)^3\right], & |r_i| \leq k \\ \frac{k^2}{6}, & |r_i| > k \end{cases} \tag{4}$$

where $r_i = E_i^{\hat{p}_i} - \hat{E}_i$ is the residual and k is a tuning constant that is commonly set to 4.685σ to produce 95% efficiency when the errors are normally distributed with standard deviation σ. To set the value of k, we estimate the standard deviation as $\hat{\sigma} = \text{MAR}/0.6745$. From Eq. 4, residuals with absolute values greater than k are considered outliers and are rejected for gradient computation. In each gradient step, we sample from both DFT- and EIP-labeled configurations to form a batch for gradient back-propagation, and the MAR is estimated on residuals of both DFT- and EIP-labeled configurations in the batch. Since the MAR and $\hat{\sigma}$ are dynamically estimated during training, the Tukey loss does not introduce additional hyperparameters.

4.3 Multi-task Pretraining

We propose a multi-task pretraining strategy to encode the domain knowledge in physics-based EIPs to the parameters of the representation learning module by jointly predicting the set of EIP calculations for configurations with only EIP predictions, $\mathcal{C}_{\text{EIP}}$, and optimizing the following multi-task regression loss:

$$\mathcal{L} = \frac{1}{n|\mathcal{P}|} \sum_{p \in \mathcal{P}} \sum_{i=m+1}^{m+n} (\hat{E}_i^p - E_i^p)^2.$$

During pretraining, we couple the representation learning module with $|\mathcal{P}|$ prediction modules (MLPs) to generate predictions corresponding to different physics-based EIPs, i.e., $\hat{E}_i^p = f_{\text{pred}}^p \circ f_{\text{rep}}(C_i)$. The pretrained representation module is then fine-tuned with a randomly initialized prediction module for the downstream DFT prediction task. Note that the representation module could either be naively fine-tuned on configurations with DFT energies by optimizing an MSE loss, or on the training set generated by our proposed label augmentation method by optimizing Eq. 3.

Although transfer learning has been successful in various application domains, it could easily hinder model performance on the target task if the pretraining tasks are unrelated to the target task (negative transfer). We argue that predicting the output of physics-based EIPs is relevant and beneficial to the target task of predicting DFT energies. Although physics-based EIPs are not perfectly accurate across the space of all possible spatial arrangements of atoms, their functional forms incorporate prior physical information that allow them to correlate with DFT over this space. In Sec. 5, we empirically demonstrate that the multi-task pretraining strategy successfully encodes domain knowledge into configuration representations and creates a smoother DFT energy surface.

4.4 Combining Label Augmentation and Multi-task Pretraining

The label augmentation and multi-task pretraining methods outlined above can be combined with relative ease. The procedure is similar to the ordinary label augmentation strategy, but rather than using a randomly initialized representation module f_{rep} for the final NN training, the representation module produced by the multi-task pretraining method is used during fine-tuning. Fig. 2 provides a schematic overview of both strategies, how they relate to one another, and how they can be combined.

5 Experiments

To test our methodology, we experiment with three datasets: the ANI-Al [38] dataset and the KIM-Si [25] dataset each with a single species, as well as a multispecies AgAu dataset [48]. Detailed descriptions of the three datasets are provided in Appendix A.2. For each dataset, we generate three splits by randomly assigning 20% of the DFT-labeled configurations as test sets and the other 80% as training sets. During training, we use 20% of the training set as a validation set for model selection. All of the reported experimental results are averaged over three different splits to avoid over-fitting to a specific split. We release our code [2] and the KIM-Si dataset [3] for reproducing our experimental results and continuous works.

5.1 Experimental Setting

5.1.1 Neural Network-based Potentials

We evaluate our proposed strategies on two classes of neural network potentials that reflect the majority of machine learning potentials currently in use. The first represents atomic environments using pre-computed descriptors and learns non-linear transformations (MLPs) to map the descriptors to atomic embeddings, while the second uses GNNs to learn atomic representations from configuration graphs. We select one representative potential from each class. For our MLP-based potential, we use the Smooth Overlap of Atomic Positions (SOAP) [1] atomic environment descriptor together with a representation module and prediction module consisting of MLPs; we term this potential SOAPNet in later discussions. For our GNN-based potential, we select SchNet [36], CGCNN [53],

[2] `https://github.com/shuix007/EIP4NNPotentials`

[3] `https://doi.org/10.6084/m9.figshare.21266064`

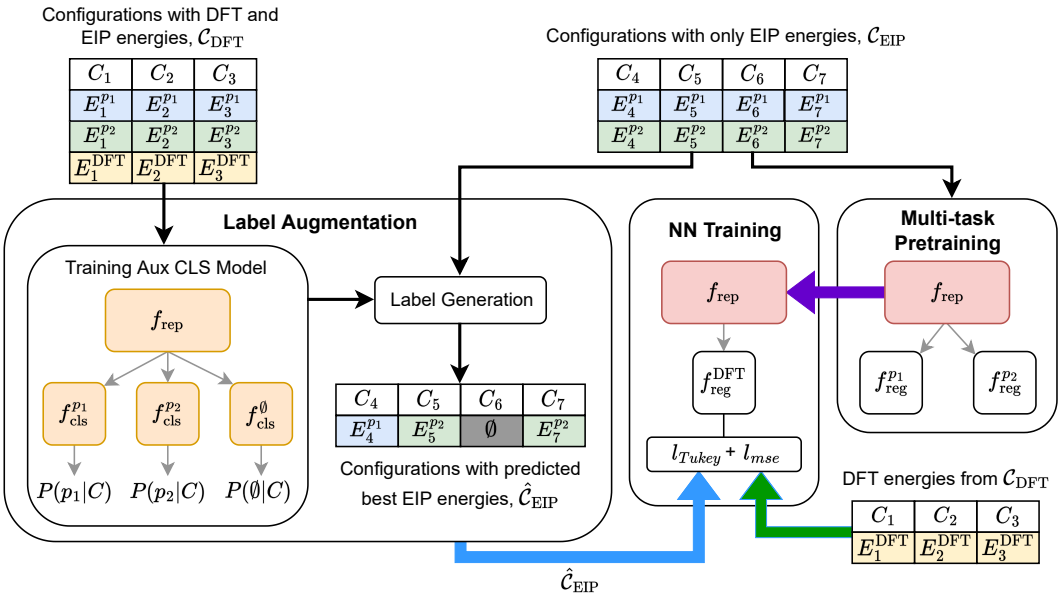


Figure 2: Illustration of the Label Augmentation (LA) and Multi-task Pretraining (MP) strategies and their usage in the training of NN-based potentials for the case of two physics-based EIPs, p_1 and p_2. In LA, a classifier is trained to predict the most accurate EIP energies $\hat{\mathcal{C}}_{\text{EIP}}$ for the unlabeled training instances, which are combined with the DFT energies from $\mathcal{C}_{\text{DFT}}$ in the loss function when training the final NN (blue arrow + green arrow). In MP, the representation module f_{rep} is pretrained by simultaneously fitting the energies of each physics-based EIP before it is inherited as the initial state for the representation module in the final NN training, where the DFT-labeled instances are again used (purple arrow + green arrow). The two strategies can be combined by incorporating $\hat{\mathcal{C}}_{\text{EIP}}$ into the loss while also using multi-task pretraining to initialize f_{rep} (blue arrow + purple arrow + green arrow).

and GemNet [16]. In our label augmentation experiments, we set the representation module of the auxiliary classification model to be the same kind as the corresponding NN potential, e.g., the classification model used for training the SchNet potential has a SchNet GNN as its representation module. Details about the model and training hyper-parameters can be found in Appendix A.4.

5.1.2 Selection of Physics-based EIPs

The physics-based EIPs used in our experiments were selected to encompass differing levels of functional complexity. Because physics-based EIPs are designed for specific elemental species (in this case, aluminum, silicon, and gold and silver systems), a different set of physics-based EIPs had to be chosen for each dataset. A total of ten physics-based EIPs were used for aluminum, eight for silicon, and two for the gold-silver system (see Tabs. 2). They were taken mainly from the Open Knowledgebase of Interatomic Models (OpenKIM)[4] repository. [41, 42]. For detailed information, see Appendix A.3.

5.2 Experimental Results

5.2.1 Performance of Label Augmentation and Multi-task Pretraining

As shown in Tab. 1, our proposed strategies improve the performance of the four baseline NNs on the three benchmark datasets. In particular, the label augmentation strategy improves the baseline NNs by 5% to 51%, while the multi-task pretraining strategy improves the baselines by 2% to 55%. Combining the two strategies gives further improvement.

[4] `https://openkim.org/`

Table 1: Performance of the two proposed strategies on DFT energy prediction tasks. We report the configuration-level and atom-level mean absolute error (MAE, lower is better) in eV and eV/atom, respectively. We denote the label augmentation strategy by LA and the multi-task pretraining strategy by MP. Best performance is shown in bold. Cases where the training procedure failed due to running out of memory are marked OOM.

	KIM-Si			ANI-Al			AgAu		
	Config	Atom	Improv.	Config	Atom	Improv.	Config	Atom	Improv.
Best EIP	1.6326	0.2524	-	46.4869	0.3561	-	4.4587	0.2063	-
SOAPNet	0.7706	0.0975	-	0.2153	0.0017	-	0.5422	0.0226	-
+LA	0.5595	0.0704	27.58%	0.1786	0.0014	18.14%	0.5067	0.0205	07.92%
+MP	0.5717	0.0733	25.28%	0.1744	0.0014	19.12%	0.3962	0.0154	29.34%
+MP+LA	**0.5307**	**0.0657**	**31.88%**	**0.1697**	**0.0013**	**22.13%**	**0.3858**	**0.0154**	**30.23%**
SchNet	0.4805	0.0718	-	0.1693	0.0014	-	0.7290	0.0290	-
+LA	0.4015	0.0549	19.99%	0.0845	0.0007	51.24%	0.6815	0.0266	07.33%
+MP	0.4034	0.0569	18.40%	0.1296	0.0010	26.00%	**0.3353**	**0.0130**	**54.65%**
+MP+LA	**0.3719**	**0.0490**	**27.17%**	**0.0816**	**0.0006**	**53.27%**	0.3496	0.0135	52.80%
CGCNN	0.9314	0.1410	-	0.2410	0.0019	-	1.6683	0.0625	-
+LA	0.7476	0.1050	22.61%	0.1786	0.0014	25.44%	1.6065	0.0589	04.71%
+MP	0.8457	0.1253	10.16%	0.2206	0.0017	07.80%	1.4377	0.0532	14.38%
+MP+LA	**0.7435**	**0.1005**	**24.44%**	**0.1392**	**0.0011**	**41.65%**	**1.3857**	**0.0499**	**18.55%**
GemNet	0.5138	0.0546	-	OOM	OOM	OOM	0.9257	0.0342	-
+LA	0.4691	0.0511	07.55%	OOM	OOM	OOM	0.8381	0.0300	10.87%
+MP	0.5024	0.0531	02.48%	OOM	OOM	OOM	**0.5057**	**0.0185**	**45.71%**
+MP+LA	**0.4651**	**0.0476**	**11.12%**	OOM	OOM	OOM	0.6074	0.0218	35.30%

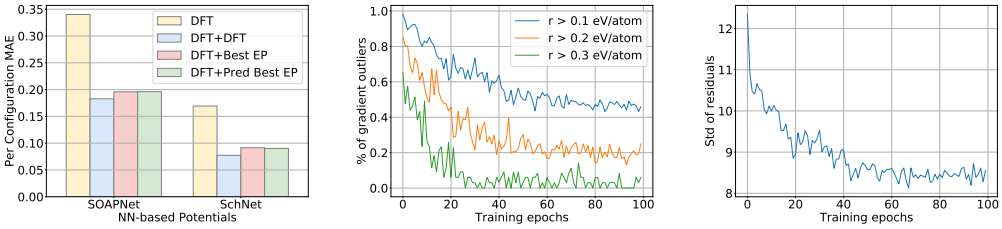
5.2.2 EIP Energies as High-Quality Supervision Signals for Training NN Potentials

Recall that in the label augmentation strategy, an auxiliary classification model selects unlabeled configurations and predicts their corresponding best-performing physics-based EIPs, which are subsequently used to label them for the training of the NN potential. These configurations are then labeled by the predicted best-performing physics-based EIPs for NN potential training. We investigate the quality of EIP labels and the auxiliary classification model by training NN potentials on three augmented training sets in which the selected unlabeled configurations are labeled by three different sources: DFT-based energies, ground-truth best-performing-EIP energies, and predicted best-performing EIP energies. The DFT-labeled configurations (0.8K) are the same for the three training sets. We only conduct this experiment on the ANI-Al dataset, as all of its configurations have DFT energies available. The number of selected unlabeled configurations is shown in Tab. 6.

Fig. 3a shows the performance of the NN potentials trained on the three augmented training sets and the original training set (where only DFT-labeled configurations are used). As shown in the figure, expanding the training set with ground truth DFT calculations (blue) greatly improves the baseline MAE (yellow bar, model trained on the original training set). Labeling configurations with the ground-truth best-performing physics-based EIPs (red bar) performs slightly worse than with DFT energies (blue bar) but still much better than the baseline (yellow bar). This demonstrates that physics-based EIPs are valuable sources of supervision signals for training NN-based potentials. Using the predicted best-performing physics-based EIPs for labeling (green bar) performs on par with the ground-truth best-performing physics-based EIP labeling (red), revealing the utility of the auxiliary classification model.

5.2.3 Importance of the Robust Tukey Loss

We next conduct experiments to investigate the importance of the Tukey loss and its ability to reject outliers during training. As before, we only conduct this experiment on the ANI-Al dataset, using DFT energies for unlabeled configurations to determine noise and outliers introduced by the predicted best-performing physics-based EIPs. We define an unlabeled configuration with a predicted best-performing physics-based EIP to be an outlier if the absolute difference between its physics-based EIP energy and its DFT energy is larger than a threshold, i.e., $|r_i| = \frac{1}{N_i}|E_i^{\text{DFT}} - E_i^{\hat{p}_i}| > c$ where N_i



(a) (b) (c)

Figure 3: (a) Performance of NN potentials trained on the original training sets (yellow bar, DFT-labeled configurations only) and three augmented training sets whose unlabeled configurations are labeled by DFT energies (blue bar), ground-truth best-performing physics-based EIP energies (red bar), and predicted best-performing physics-based EIP energies (green bar). (b) Percentage of outliers used for computing gradient during training. (c) Standard deviation of residuals during training.

is the number of atoms in configuration i. We categorize the outliers as mild, normal, and severe by setting c to $\{0.1, 0.2, 0.3\}$. The initial number of outliers introduced by the unlabeled configurations can be found in Tab. 6. Figs. 3b and 3c show that the number of outliers of all kinds included for computing gradients decreases as the training proceeds, demonstrating that as the NN potential gets progressively more accurate, the Tukey loss can effectively eliminate outliers from the training set. We also conduct an ablation study by replacing the Tukey loss in Eq. 3 with the MSE loss. The results in Tab. 4 show that the models' performance degrades without the Tukey loss.

5.2.4 Visualization of Pretrained Configuration Representations

Fig. 4 plots the t-SNE [44] 2D projections of the training silicon configuration representations colored by their per-atom DFT energies. The left-hand figure plots representations generated by a SchNet with random weights and the right-hand figure plots representations generated by a SchNet pretrained by the multi-task strategy. The representations generated by the randomly initialized SchNet do not exhibit any clear patterns and the energy surface is rough. In the right-hand figure, representations of the atomic cluster configurations (i.e., isolated groups of atoms) and the bulk configurations (crystals) are clearly separated and form clusters in the t-SNE 2D space. The per-atom DFT energy surface of the right-hand figure is much smoother, i.e., configurations with similar energies are close to one another after pretraining. This verifies our previous statement that, although physics-based EIPs lack complete generalizability, they nonetheless correlate reasonably well with DFT over atomic configuration space, and demonstrates that our proposed multi-task pretraining strategy successfully encodes domain knowledge into the NN-based potentials. T-SNE plots for SOAPNet on both datasets and SchNet on the ANI-Al dataset show similar patterns. These figures may be found in the Appendix A.5.

6 Conclusion

In this paper, we show that physics-based EIPs can be used to incorporate domain knowledge into machine learning EIPs by providing additional supervision signals on a large body of unlabeled training instances. Two generic strategies are formulated. The first, label augmentation, is based on weakly supervised learning and uses a classifier to select unlabeled training instances to supplement the original labeled training set. The second, multi-task pretraining, is based on transfer learning and uses the predictions of a set of physics-based EIPs on unlabeled training instances to pretrain a machine learning EIP that is subsequently fine-tuned on the original training set. Our methodology is proven using several experiments, including a cross-validation study that demonstrates significant performance gains for either strategy individually and further improvement when they are combined, a label augmentation study that shows that using EIP energies as surrogate labels provides nearly as much robustness as using additional DFT-labeled training instances, and a statistical experiment that illustrates the utility of the Tukey loss in diminishing noise and excluding outliers. Finally, we present reduced-dimension visualizations that indicate the multi-task pretraining strategy yields training set feature vectors over which the energy varies smoothly.

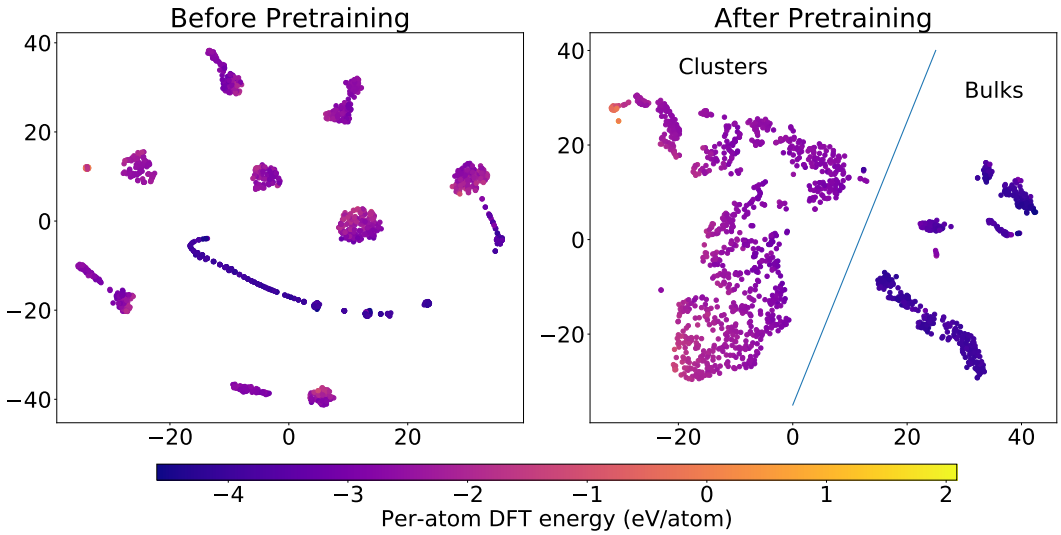


Figure 4: T-SNE plots of silicon configuration representations generated by a randomly initialized SchNet (left) and a SchNet pretrained with our proposed multi-task pretraining strategy (right). Configurations are colored by their per-atom DFT energies. Representations generated by the pretrained SchNet naturally form two clusters that correspond to the atomic cluster configurations (i.e., isolated groups of atoms) and the bulk configurations (i.e., crystals).

One direction for future work is to replace the heuristic EIPs by more accurate but still computationally cheap methods, such as the semi-empirical tight binding method. These methods better approximate DFT and thus can lead to further improvement when combined with our proposed approaches.

7 Acknowledgement

This work was supported in part by NSF (1447788, 1704074, 1757916, 1834251, 1834332), Army Research Office (W911NF1810344), the startup funds from the Presidential Frontier Faculty Program at the University of Houston, Intel Corp, and Amazon Web Services. Access to research and computing facilities was provided by the Minnesota Supercomputing Institute. We thank the anonymous reviewers for their feedback during NeurIPS 2022 review process. We are grateful to Sijie He and Yingxue Zhou for their insightful discussion and inspiration.

References

[1] Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Phys. Rev. B*, 87:184115, May 2013. doi: 10.1103/PhysRevB.87.184115. URL `https://link.aps.org/doi/10.1103/PhysRevB.87.184115`.

[2] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1): 1–11, 2022.

[3] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[4] Jörg Behler. Four generations of high-dimensional neural network potentials. *Chemical Reviews*, 121(16):10037–10072, 2021.

[5] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2838, 2015.

[6] Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International journal of computer vision*, 19(1):57–91, 1996.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] 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 Catalysis*, 2021. doi: 10.1021/acscatal.0c04525.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[12] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[13] John Fox and Sanford Weisberg. Robust regression. *An R and S-Plus companion to applied regression*, 91, 2002.

[14] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[15] Wei Gao, Lu Wang, Zhi-Hua Zhou, et al. Risk minimization in the presence of label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[16] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6790–6802. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/35cf8659cfcb13224cbd47863a34fc58-Paper.pdf`.

[17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[18] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[20] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, 2018.

[21] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2019.

[22] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867, 2020.

[23] Peter J Huber. Robust statistics. In *International encyclopedia of statistical science*, pages 1248–1251. Springer, 2011.

[24] Maria Kalantzi and George Karypis. Position-based hash embeddings for scaling graph neural networks. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 779–789. IEEE, 2021.

[25] Daniel S Karls. *Transferability of empirical potentials and the Knowledgebase of Interatomic Models (KIM)*. PhD thesis, University of Minnesota, 2016.

[26] Efthimios Kaxiras. *Atomic and Electronic Structure of Solids*. Cambridge University Press, 2003. doi: 10.1017/CBO9780511755545.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[29] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[30] Mufei Li, Jinjing Zhou, Jiajing Hu, Wenxuan Fan, Yangkang Zhang, Yaxin Gu, and George Karypis. Dgl-lifesci: An open-source toolkit for deep learning on graphs in life science. *ACS omega*, 6(41):27233–27238, 2021.

[31] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[32] Saurav Manchanda, Mohit Sharma, and George Karypis. Distant-supervised slot-filling for e-commerce queries. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 677–686, 2021. doi: 10.1109/BigData52589.2021.9671825.

[33] Costas Mavromatis and George Karypis. Graph infoclust: Maximizing coarse-grain mutual information in graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 541–553. Springer, 2021.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[35] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2020.

[36] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[37] Zeren Shui and George Karypis. Heterogeneous molecular graph neural networks for predicting molecule properties. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 492–500. IEEE, 2020.

[38] Justin S Smith, Benjamin Nebgen, Nithin Mathew, Jie Chen, Nicholas Lubbers, Leonid Burakovsky, Sergei Tretiak, Hai Ah Nam, Timothy Germann, Saryu Fensin, et al. Automated discovery of a robust interatomic potential for aluminum. *Nature communications*, 12(1):1–13, 2021. doi: 10.1038/s41467-021-21376-0.

[39] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[40] Ellad B Tadmor and Ronald E Miller. *Modeling materials: continuum, atomistic and multiscale techniques*. Cambridge University Press, 2011.

[41] Ellad B. Tadmor, Ryan S. Elliott, James P. Sethna, Ronald E. Miller, and Chandler A. Becker. The potential of atomistic simulations and the Knowledgebase of Interatomic Models. *JOM*, 63 (7):17, July 2011. doi: 10.1007/s11837-011-0102-6. URL `http://dx.doi.org/10.1007/s11837-011-0102-6`.

[42] Ellad B. Tadmor, Ryan S. Elliott, Simon R. Phillpot, and Susan B. Sinnott. NSF cyberinfrastructures: A new paradigm for advancing materials simulation. *COSSMS*, 17(6):298–304, December 2013. doi: 10.1016/j.cossms.2013.10.004. URL `http://dx.doi.org/10.1016/j.cossms.2013.10.004`.

[43] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[45] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 839–847, 2017.

[46] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[47] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, et al. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

[48] YiNan Wang, LinFeng Zhang, Ben Xu, XiaoYang Wang, and Han Wang. A generalizable machine learning potential of ag–au nanoalloys and its application to surface reconstruction, segregation and diffusion. *Modelling and Simulation in Materials Science and Engineering*, 30 (2):025003, 2021.

[49] Mingjian Wen and Ellad B Tadmor. Uncertainty quantification in molecular simulations with dropout neural network potentials. *npj Comput. Mater.*, 6(1):124, 2020. doi: 10.1038/s41524-020-00390-8.

[50] Mingjian Wen, Stephen Carr, Shiang Fang, Efthimios Kaxiras, and Ellad B. Tadmor. Dihedral-angle-corrected registry-dependent interlayer potential for multilayer graphene structures. *Phys. Rev. B*, 98(23):235404, dec 2018. doi: 10.1103/physrevb.98.235404.

[51] Mingjian Wen, Yaser Afshar, Ryan S Elliott, and Ellad B Tadmor. KLIFF: A framework to develop physics-based and machine learning interatomic potentials. *Computer Physics Communications*, 272:108218, 2022.

[52] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

[53] Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120:145301, Apr 2018. doi: 10.1103/PhysRevLett.120.145301. URL `https://link.aps.org/doi/10.1103/PhysRevLett.120.145301`.

[54] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

[55] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical review letters*, 120(14):143001, 2018.

[56] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[57] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5 (1):44–53, 2018.

[58] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We included in detail experimental settings, code repo, and datasets in Section 5.1 and Appendix A.4 which is sufficient for reproducing the main experimental results.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5.1 and Appendix A.4.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix A.5.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section A.4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] We appropriately cited the source of datasets that we used in the paper, and open source software we used in the experiments.

   (b) Did you mention the license of the assets? [N/A]

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] One of the datasets we used is open-source while the other we curated ourselves.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Periodic Boundary Conditions

Under periodic boundary conditions (PBCs), the positions of atoms outside the simulation cell are obtained by generating periodic images of those within the cell through translations commensurate with its periodicity. This methodology is capable of modeling infinite systems because the interactions between atoms separated by more than a modest cutoff distance are very small and thus ignored when defining empirical models. This limited range of interaction gives rise to the concept of an *atomic environment*. The environment of a given atom consists of itself and all other atoms, including periodic images, that fall within a prescribed cutoff distance of it. The consequence of this locality is that an infinite system can be modeled exactly using a finite periodic cell so long as a sufficient number of periodic images surrounding it are explicitly accounted for. An example of PBCs for a two-dimensional square cell and a local atomic environment is illustrated in Fig. 5.

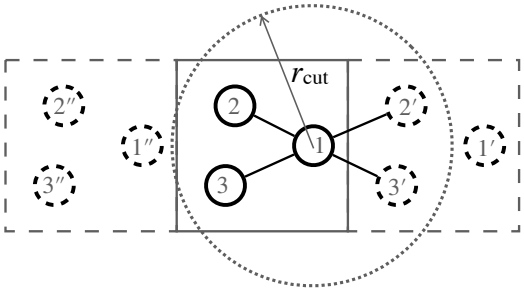


Figure 5: Illustration of periodic boundary conditions for a two-dimensional simulation cell (solid square) containing three atoms. For simplicity, only periodic images in the horizontal direction are shown. The local environment of atom 1 (dashed circle) contains all atoms and their periodic images that fall within a prescribed cutoff distance r_{cut} of it.

A.2 Datasets

The ANI-Al dataset [38] consists of DFT energies of 6,352 configurations of aluminum in liquid, solid, and liquid-solid coexistence phases, each containing up to 250 atoms.

The KIM-Si dataset is a new dataset (soon to be published) that we generated for silicon comprising a total of 16,110 configurations. It builds upon the training set first used in [25] and contains a total of 14,510 perturbed bulk structures, 1,525 randomly generated atomic clusters ranging in size from two atoms to ten atoms, 3 ideal surface defects, and 72 nanostructures constructed by composing two-dimensional structures of silicon in the graphene and silicene geometries. In total, for the KIM-Si dataset, DFT energies are available for 510 of the bulk configurations and the 1600 non-bulk configurations.

The AgAu dataset [48] is developed to study the catalytic properties of AgAu binary nanoalloys. It consists of configurations for elemental Ag and Au, as well as AgAu binary alloy. The dataset is generated using an active learning strategy. First, an initial dataset of Ag, Au, and AgAu in body-centered cubic (BCC), face-centered cubic (FCC), and hexagonal close-packed (HCP) structures are labelled using DFT. Then, an ensemble of models are trained on the data and new configurations are selected to be further labelled by DFT based on the uncertainty obtained from the ensemble. This process is iterated multiple times. In this work, we randomly select 10k of the AuAg binary configurations to test our methodology on multispecies systems.

A.3 Selection of Physics-based EIPs

We show the selected EIPs used in our experiments and their accuracy in Table 2 for reference. From the Table, we can observe that the physics-based EIPs used in label augmentation and multi-task pretraining perform significantly worse than the NN-based potentials and our proposed strategies. This is mainly because the functional form of an EIP is usually developed for specific configurations and cannot transfer very well to arbitrary structures. For example, the Tersoff (T3) potential employed

on the KIM-Si dataset was developed for cubic diamond crystal structures and thus will not perform well for other structures in the test set.

Table 2: EIPs used in experiments and their accuracy. Standard deviations are shown in parentheses.

Species	EIP name	Config	Atom	Link
Si	EDIP	4.3063(0.1767)	0.6071(0.0187)	https://doi.org/10.25950/545ca247
	MEAM	3.7315(0.0603)	0.5235(0.0055)	https://doi.org/10.25950/b8dc8b23
	SW (BalamaneHalicioglu Tiller)	21.3235(0.6292)	2.7169(0.0765)	https://doi.org/10.25950/3dc2cb7f
	SW (ZhangXieHu)	17.6129(0.4513)	2.2711(0.0417)	https://doi.org/10.25950/32a4bf2c
	Tersoff (T2)	1.6326(0.1262)	0.2524(0.0177)	https://doi.org/10.25950/cadc4e78
	Tersoff (T3)	11.8825(0.3287)	1.5889(0.0449)	https://doi.org/10.25950/d6e8a23e
	Tersoff (ErhartAlbe)	6.4907(0.2466)	0.8393(0.0279)	https://doi.org/10.25950/6aa22835
	Tersoff (TMOD)	5.4128(0.2036)	0.7210(0.0259)	https://doi.org/10.1016/j.commatsci.2006.07.013
Al	EMT	50.8366(0.8829)	0.3922(0.0053)	https://doi.org/10.25950/bdbaee6a
	Morse (LowCutoff)	143.3241(1.4645)	1.1433(0.0082)	https://doi.org/10.25950/977dc2ac
	Morse (MedCutoff)	120.6183(1.4162)	0.9711(0.0080)	https://doi.org/10.25950/474ccb33
	Morse (HighCutoff)	116.5143(1.4021)	0.9400(0.0079)	https://doi.org/10.25950/45d9848f
	EAM (ErcolessiAdams)	78.8916(1.7788)	0.6105(0.0097)	https://doi.org/10.25950/bc2d2486
	EAM (SturgeonLaird)	79.4206(2.0645)	0.6147(0.0115)	https://doi.org/10.25950/d62edb43
	EAM (WineyKubotaGupta)	95.8777(0.9002)	0.7587(0.0054)	https://doi.org/10.25950/23542694
	EAM (ZopeMishin)	46.4869(0.6270)	0.3561(0.0042)	https://doi.org/10.25950/26dbac6e
	EAM (Zhakhovsky)	66.0950(1.3760)	0.5091(0.0075)	https://doi.org/10.25950/c3a79c52
	EAM (ZhouJohnsonWadley)	69.1926(0.5550)	0.5289(0.0020)	https://doi.org/10.25950/c775fc98
AgAu	EAM (ZhouJohnsonWadley)	5.7077(0.0942)	0.2438(0.0077)	https://doi.org/10.25950/d77528cf
	EMT (JacobsenStoltzeNorskov)	4.4587(0.1436)	0.2063(0.0012)	https://doi.org/10.25950/485ab326

Most of the silicon physics-based EIPs were trained using atomic configurations similar to the bulk diamond configurations in our dataset. This raises the concern of information leakage, where the NN potentials indirectly learn information about the test set during training through supervision signals provided by the physics-based EIPs, since they were potentially trained on part of the test set. However, in the first two folds of our cross-validation, the perturbed diamond configurations constitute only 2.4% of the test set, and in the third fold constitute only 3.6%. In the case of aluminum, none of the physics-based EIPs used in our experiments were fitted to any configurations similar to those in the ANI-Al dataset. Altogether, we conclude that the impact of this effect is minimal.

A.4 Experimental Settings

We set the number of hidden dimensions of all NNs to 128, the number of stacked NN layers in representation modules (GNNs and MLPs) to 5, and use the shifted softplus activation function in all nodes. We choose sum pooling to be the readout function and an MLP to be the prediction module. All of our models are optimized using the Adam algorithm [27] with a learning rate of 1e-3 and a batch size of 32. For the multi-task pretraining and the training of the auxiliary classification model, we use a slated triangular scheduler [20] for the initial warm up of the weights, and subsequently decrease the learning rate linearly. For training the NNs with DFT energies, we decrease the learning rate linearly from 1e-3 to 1e-5. All NNs are trained for 100 epochs. Results are reported on the checkpoint with the best validation performance. We use validation performance to select the hyperparameter α that controls the contribution of EIP-labeled configurations (cf. Section 4.2) from [0.01, 0.05, 0.1, 0.5, 1, 5, 10].

Our code for the (G)NN potentials and experiments is implemented using PyTorch [34]. The implementation of SchNet is modified from DGL [47], and DGL-LifeSci [30]. The implementation of CGCNN and GemNet are modified from PyG [12] and Open Catalyst Project [8] All experiments are conducted on a machine with an Intel(R) Core(TM) i9-10900F CPU and an Nvidia RTX-3090 GPU. Pretraining SchNet and CGCNN on the ANI-Al and the KIM-Si dataset takes 1.5hr and 0.5hr, respectively, while retraining SOAPNet on both datasets takes a few minutes. The time cost of label augmentation depends largely on the number of unlabeled configurations added to the training set. In our case, experiments in all settings take less than 2hrs to finish.

A.5 Additional Experimental Results

A.5.1 Performance of the Two Strategies with Standard Deviation

We report performance of the two strategies on an average over three runs on three random splits and report the mean/std in Tab. 3

Table 3: Performance of the two proposed strategies on DFT energy prediction tasks. We report the configuration-level and atom-level mean absolute error (MAE, lower is better) in eV and eV/atom, respectively. We denote the label augmentation strategy by LA and the multi-task pretraining strategy by MP. Standard deviations are shown in parentheses. Cases where the training procedure failed due to running out of memory are marked OOM.

	KIM-Si		ANI-Al		AgAu	
	Config	Atom	Config	Atom	Config	Atom
SOAPNet	0.7706 (0.0658)	0.0975 (0.0012)	0.2153 (0.0228)	0.0017 (0.0002)	0.5422 (0.0305)	0.0226 (0.0016)
+LA	0.5595 (0.0336)	0.0704 (0.0026)	0.1786 (0.0239)	0.0014 (0.0002)	0.5067 (0.0470)	0.0205 (0.0021)
+MP	0.5717 (0.0232)	0.0733 (0.0038)	0.1744 (0.0249)	0.0014 (0.0002)	0.3962 (0.0446)	0.0154 (0.0019)
+MP+LA	0.5307 (0.0375)	0.0657 (0.0033)	0.1697 (0.0193)	0.0013 (0.0002)	0.3858 (0.0398)	0.0154 (0.0022)
SchNet	0.4805 (0.0345)	0.0718 (0.0096)	0.1693 (0.0106)	0.0014 (0.0001)	0.7290 (0.0692)	0.0290 (0.0026)
+LA	0.4015 (0.0687)	0.0549 (0.0042)	0.0845 (0.0095)	0.0007 (0.0001)	0.6815 (0.0659)	0.0266 (0.0026)
+MP	0.4034 (0.0723)	0.0569 (0.0115)	0.1296 (0.0164)	0.0010 (0.0001)	0.3353 (0.0331)	0.0130 (0.0012)
+MP+LA	0.3719 (0.0706)	0.0490 (0.0059)	0.0816 (0.0064)	0.0006 (0.0001)	0.3496 (0.0295)	0.0135 (0.0011)
CGCNN	0.9314 (0.0379)	0.1410 (0.0099)	0.2410 (0.0318)	0.0019 (0.0002)	1.6683 (0.0851)	0.0625 (0.0049)
+LA	0.7476 (0.0516)	0.1050 (0.0036)	0.1786 (0.0241)	0.0014 (0.0002)	1.6065 (0.1314)	0.0589 (0.0055)
+MP	0.8457 (0.0611)	0.1253 (0.0063)	0.2206 (0.0380)	0.0017 (0.0003)	1.4377 (0.0852)	0.0532 (0.0036)
+MP+LA	0.7435 (0.0510)	0.1005 (0.0043)	0.1392 (0.0213)	0.0011 (0.0002)	1.3857 (0.1053)	0.0499 (0.0044)
GemNet	0.5138 (0.1097)	0.0546 (0.0023)	OOM	OOM	0.9257 (0.1133)	0.0342 (0.0045)
+LA	0.5024 (0.1225)	0.0531 (0.0016)	OOM	OOM	0.5057 (0.0691)	0.0185 (0.0026)
+MP	0.4691 (0.1296)	0.0511 (0.0075)	OOM	OOM	0.8381 (0.0567)	0.0300 (0.0023)
+MP+LA	0.4651 (0.1410)	0.0476 (0.0041)	OOM	OOM	0.6074 (0.0499)	0.0218 (0.0022)

Table 4: Configuration-level MAE (eV) with and without the Tukey loss.

	KIM-Si		ANI-Al	
	SOAPNet	SchNet	SOAPNet	SchNet
w/o Tukey	0.5556	0.4374	0.1906	0.1031
w/ Tukey	0.5595	0.4015	0.1786	0.0845

A.5.2 Ablation Study of the Tukey Loss

Tab. 4 shows the configuration-level MAE for the KIM-Si and ANI-Al datasets with and without using the Tukey loss.The results show that the models' performance degrades without the Tukey loss.

A.5.3 Effect of Expanding Training Set with Different Confidence Level

We expand the training set with DFT energies with EIP-labeled configurations with different confidence level computed by the auxiliary classification model, i.e., $P(\hat{p} \mid C)$ (see Section 4.2). We first sort the configurations by their confidence scores and assign them to different confidence groups. Configurations with a confidence score higher than the 0.66 quantile are assigned to high confidence group, configurations with a confidence score lower than the 0.33 quantile are assigned to low confidence group, other configurations are assigned to the medium confidence group. We expand the DFT-labeled configurations to three different training sets by adding configurations belonging to different groups. Results are shown in Tab. 5. The results suggest that the unlabeled configurations with medium confidence contribute the most to the training while configurations with high confidence contribute the least. This is because the high confidence configurations may be very similar to the configurations in the training set and thus do not provide extra information. Low and medium confidence configurations are more dissimilar than those in the original training set and can provide more information.

A.5.4 Number of Outliers and Configurations Selected By the Classification Model

We show the number of configurations and outliers introduced by the auxiliary classification model in Tab. 6. Results in the table show that the auxiliary classification model reaches an accuracy of 75% to 78% for screening configurations and selecting reasonable EIPs for labeling them. This demonstrates the utility of the classification model.

Table 5: MAE (eV and eV/atom) on training sets expanded by configurations with different confidence level.

		KIM-Si		ANI-Al	
	Confidence	Config	Atom	Config	Atom
SOAPNet	Low	0.5793	0.0732	0.2174	0.0017
	Medium	0.5728	0.0725	0.2133	0.0017
	High	0.6196	0.0778	0.2361	0.0019
SchNet	Low	0.4817	0.0657	0.1140	0.0009
	Medium	0.4916	0.0668	0.1130	0.0009
	High	0.5004	0.0740	0.1242	0.0010

Table 6: Average number of configurations and outliers selected by the classification models on the ANI-Al dataset.

	#Mild	#Normal	#Severe	#Selected	#Unlabeled
SOAPNet	226	42	5	1211	5081
SchNet	271	50	9	1300	5081

A.5.5 Influence of the Selected Potentials

We conducted an ablation study on the KIM-Si dataset to investigate the influence of the selected set of EIPs on our two proposed strategies. Among the eight EIPs for Si, we selected two EIPs that work the best and the worst on the KIM-Si dataset, respectively. We apply LA and MP with the two EIPs separately and jointly (for a total of three experiments) on SOAPNet and SchNet and report their results in Tab. 7. From the table we can observe that, label augmentation (LA) with a good EIP improves the baseline performance while LA with a bad EIP does not hurt the performance very much thanks to the auxiliary classification model and the robust Tukey loss. When applying LA on a mixture of good and bad EIPs, our strategy is able to select good from bad and provide performance boost. Moreover, the more (good) EIPs leveraged in LA, the better performance it can give. For multi-task pretraining, we observe that the number and quality of the EIPs do not influence the performance by too much.

A.5.6 Visualization of Pretrained Configuration Representations

We show the t-SNE 2D projection plots for all datasets and NN potentials in Figs. 6, 7, and 8. These plots show that the multi-task pretraining successfully injects domain knowledge into the representation module of NN potentials and creates a smoother DFT energy surface.

Table 7: MAE (eV and eV/atom) of different selected set of EIPs on Si. Numbers in the brackets indicate the number of EIPs used for the experiments.

		SOAPNet		SchNet	
		Config	Atom	Config	Atom
	Baseline	0.7582	0.0987	0.5021	0.0745
+LA	Default (8)	0.5673	0.0725	0.4114	0.0545
	Best EIP (1)	0.6084	0.0797	0.4395	0.0614
	Worst EIP (1)	0.766	0.0995	0.5245	0.0782
	Mix (2)	0.5878	0.0767	0.4286	0.0596
+MP	Default (8)	0.5679	0.0741	0.4217	0.0545
	Best EIP (1)	0.6897	0.0892	0.4127	0.0561
	Worst EIP (1)	0.6734	0.0881	0.4116	0.0559
	Mix (2)	0.6549	0.0856	0.3659	0.0476

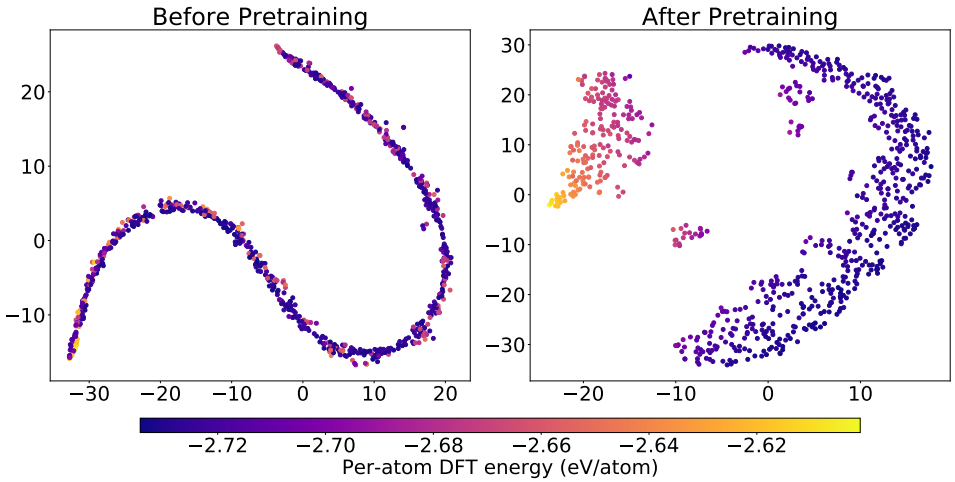


Figure 6: T-SNE plots of aluminum configuration representations generated by a randomly initialized SchNet (left) and a SchNet pretrained with our proposed multi-task pretraining strategy (right).

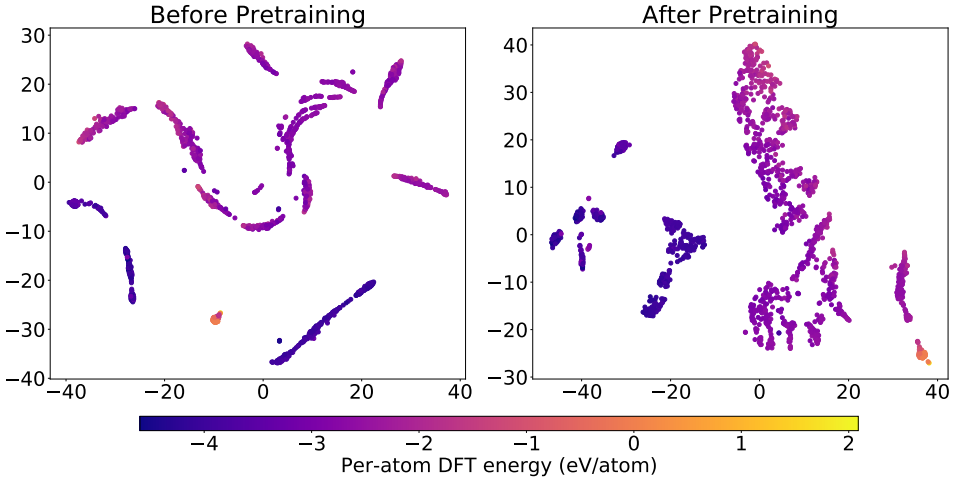


Figure 7: T-SNE plots of silicon configuration representations generated by a randomly initialized SOAPNet (left) and a SOAPNet pretrained with our proposed multi-task pretraining strategy (right).

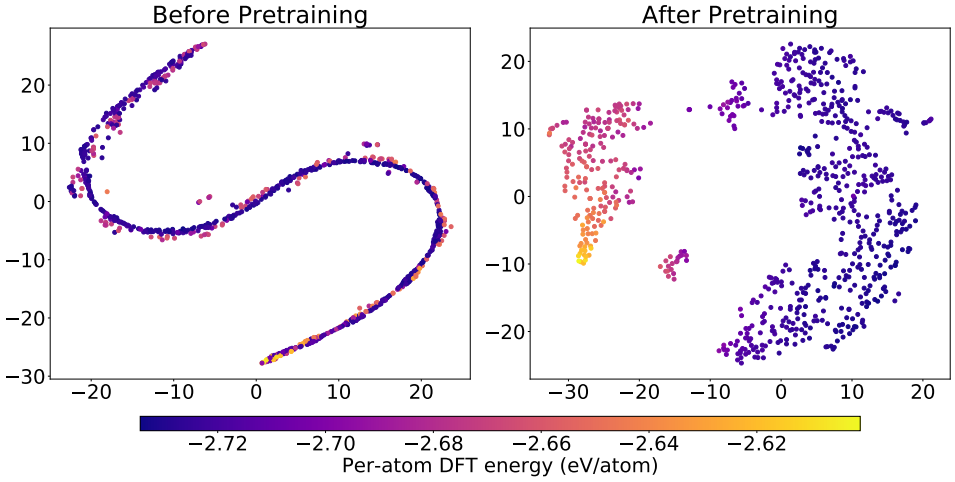


Figure 8: T-SNE plots of aluminum configuration representations generated by a randomly initialized SOAPNet (left) and a SOAPNet pretrained with our proposed multi-task pretraining strategy (right).