Published in partnership with the Shanghai Institute of Ceramics of the Chinese Academy of Sciences

https://doi.org/10.1038/s41524-024-01493-2

DPA-2: a large atomic model as a multi-task learner

Check for updates

Duo Zhang[1,2,3,30], Xinzijian Liu[1,2,30], Xiangyu Zhang[4,5], Chengqian Zhang[2,6], Chun Cai[1,2], Hangrui Bi[1,2], Yiming Du[4,5], Xuejian Qin[7,8], Anyang Peng[1], Jiameng Huang[2,9], Bowen Li[10], Yifan Shan[7,8], Jinzhe Zeng[11], Yuzhi Zhang[2], Siyuan Liu[2], Yifan Li[12], Junhan Chang[2,13], Xinyan Wang[2], Shuo Zhou[2,14], Jianchuan Liu[15], Xiaoshan Luo[16,17], Zhenyu Wang[17,18], Wanrun Jiang[1], Jing Wu[19], Yudi Yang[19], Jiyuan Yang[19], Manyi Yang[20], Fu-Qiang Gong[21], Linshuang Zhang[2], Mengchao Shi[2], Fu-Zhi Dai[1], Darrin M. York[11], Shi Liu[19,22], Tong Zhu[10,23,24], Zhicheng Zhong[7,8], Jian Lv[17], Jun Cheng[21,25,26], Weile Jia[4], Mohan Chen[1,6], Guolin Ke[2], Weinan E[1,27,28], Linfeng Zhang[1,2] ✉ & Han Wang[6,29] ✉

The rapid advancements in artificial intelligence (AI) are catalyzing transformative changes in atomic modeling, simulation, and design. AI-driven potential energy models have demonstrated the capability to conduct large-scale, long-duration simulations with the accuracy of ab initio electronic structure methods. However, the model generation process remains a bottleneck for large-scale applications. We propose a shift towards a model-centric ecosystem, wherein a large atomic model (LAM), pre-trained across multiple disciplines, can be efficiently fine-tuned and distilled for various downstream tasks, thereby establishing a new framework for molecular modeling. In this study, we introduce the DPA-2 architecture as a prototype for LAMs. Pre-trained on a diverse array of chemical and materials systems using a multi-task approach, DPA-2 demonstrates superior generalization capabilities across multiple downstream tasks compared to the traditional single-task pre-training and fine-tuning methodologies. Our approach sets the stage for the development and broad application of LAMs in molecular and materials simulation research.

An accurate interatomic potential energy surface (PES) is crucial for molecular modeling and simulations. Quantum mechanical (QM) methods, such as density functional theory (DFT)[1,2], provide satisfactory accuracy in most applications. However, their computational complexity typically scales as the cubic order of the system size, thus limiting large-scale simulations. In contrast, empirical force fields (EFF) are way more efficient, but their accuracy is often deemed insufficient for various applications. Machine learning potentials (MLPs) have emerged as a powerful approach to modeling complex materials and molecules, bridging the gap between the high accuracy of QM methods and the computational efficiency of EFFs. This has enabled the study of large-scale molecular systems with QM-level accuracy across diverse applications, including drug discovery[3,4], materials design[5–7], and catalysis[8,9], etc.

In most MLP applications, the training data is generated from scratch either through brute force ab initio molecular dynamics (MD) simulations[10] or by using a concurrent learning (or active learning) scheme capable of automatically generating the most critical data for building uniformly accurate models[11–14].

In any case, DFT-calculated energies and forces are required for each configuration in the training dataset, resulting in a substantial amount of effort spent on constructing DFT-labeled datasets. For instance, in the AlMgCu general-purpose ternary alloy MLP[15], more than 10 million CPU hours were spent on labeling the 141K training data points. Furthermore, MLPs often struggle to generalize to applications not covered by the training data[5], such as when additional elements are included in materials design or when crystal structures in a broader range of thermodynamic conditions need to be explored.

To further extend the application range of MLPs, efforts have been made to develop "universal" or "fundamental" models[16–21], referred to as large atomic models (LAMs), based on extensive density functional theory (DFT)-labeled datasets. However, the technical approach still requires further exploration, and a LAM-centric ecosystem remains to be established. The primary factors influencing this exploration process are the methods employed for model training and their subsequent application in various tasks.

During the model training stage, a single-task-based training strategy, i.e., training using consistently labeled data, remains dominant. Models generated in this way are typically expected to be directly applicable to downstream tasks in which the explored configurations are effectively

A full list of affiliations appears at the end of the paper. ✉e-mail: linfeng.zhang.zlf@gmail.com; wang_han@iapcm.ac.cn

covered by the training data. Some examples include models such as M3GNet[17], CHGNet[19], and MACE-MP-0[20], which are all trained on snapshots from DFT relaxations of the Material Project[22] structures, with M3GNet utilizing 88 K configurations across 89 chemical species and both CHGNet and MACE-MP-0 being trained on 1.58 M inorganic crystal frames from the concurrently introduced MPtrj dataset[19]; GNoME[21], trained on a dataset of inorganic crystals also starting from MP, but nearly two orders of magnitude larger than MPtrj; PreFerred Potential (PFP), trained on approximately 9 M frames of 45 elements[16]; and ALIGNN, trained on 307 K data frames of 89 elements[18].

Several limitations exist in the single-task training strategy: (1) simultaneously training multiple datasets from different application fields is not feasible due to the variations in labeling with different DFT settings. For instance, the MPtrj dataset, labeled by DFT calculations using PBE/PBE+U[23] exchange-correlation functional and plane-wave basis, cannot be concurrently trained with the ANI-1× dataset, labeled by DFT calculations using the ωB97× hybrid functional[24] and an atomic basis set, thus little possibility is left to improve the model's generalizability on molecular applications.

(2) The requirements of downstream tasks might be difficult to satisfy. For instance, a task may require DFT accuracy at the meta-general gradient approximation (meta-GGA) level. A model trained with GGA-level DFT data would not be easily adapted to fulfill this requirement.

Multi-task pre-training, combined with various strategies for downstream tasks such as fine-tuning and distillation, has emerged as a promising alternative for the development of LAMs[25–28]. By employing the multi-task training strategy[29,30], it becomes possible to jointly pre-train models using multiple datasets labeled with different DFT settings[27,31]. During fine-tuning for downstream tasks, the model's backbone, which encodes the representation of configurational and chemical spaces, is preserved and connected to one or multiple tasks heads[32,33]. As a result, the labeling methods for pre-training and fine-tuning datasets do not need to be identical. Furthermore, the downstream tasks can involve property predictions rather than PES modeling[31]. This scheme offers significant flexibility in downstream tasks and may lead to a much better generalization ability of a LAM.

Before proceeding further, let us list the requirements of a LAM that we consider to be fundamental: (1) highly generalizable, (2) extensive and respect the translational, rotational, and permutational symmetries, (3) conservative, and (4) continuous up to second-order derivatives. A model with high generalizability implies that when trained with the same amount of data, the model can achieve high accuracy[34]. The generalizability is critical in pre-training LAMs, considering that the DFT-labeled data are expensive and sparse in the configurational and chemical spaces. By conservative, we mean that the forces (and virial tensor, for periodic systems) are calculated by the derivatives of the model-predicted total energy of the system concerning atom coordinates (and cell tensor, respectively). The conservativeness and smoothness of the model are critical for energy conservation in MD simulations and are thus a compulsory requirement for calculating dynamic properties such as diffusion coefficient, viscosity, and thermal conductivity[35]. The requirements (1–4) are physical restraints imposed on a PES, thus they are necessary (but in general not sufficient) conditions for the generalizability of the LAMs.

In this context, the primary contribution of this work is the development of DPA-2, a multi-task pre-trained model that meets all the mentioned requirements and furnishes a representation suitable for a diverse array of multi-disciplinary applications, including alloys, semiconductors, battery materials, drug molecules, and more, while exhibiting a high degree of generalization for downstream tasks. The revelation of a remarkable correspondence between the learned representations by DPA-2 and existing chemical knowledge underscores the potential of the proposed model architecture and the multi-task training scheme. Furthermore, we emphasize the importance of an open and application-oriented model evaluation system for the molecular simulation community in the era of large atomic models.

The following text discusses and summarizes relevant references pertinent to the current work. In recent years, there has been rapid development in MLP models. While it is nearly impossible to provide a comprehensive list, some notable examples include the Behler–Parrinello neural network (BPNN)[36], ANI[37], deep tensor neural networks (DTNN)[38], weighted atom-centered symmetry functions (wACSF)[39], deep potential (DP)[40–42], deep potential with attention (DPA-1)[43], and embedded atom neural network (EANN)[44]. These models employ either hand-crafted or machine-learned descriptors of atomic environments, along with deep neural networks, to approximate potential energy. Other machine learning techniques, such as kernel ridge regression, are also widely used. Examples include the Gaussian approximation potential (GAP)[45], which uses a smooth overlap of atomic positions (SOAP) measure of distance between local environments[46], the Coulomb matrix[47], and gradient-domain machine learning (GDML)[48]. Some potential energy models, such as the spectral neighbor analysis method (SNAP)[49] and the moment tensor potential (MTP)[50], utilize linear regression for fitting the potential energy surface (PES). Recently, there has been a surge in the development of equivariant graph neural networks (GNN)[51,52], with examples including SchNet[53], Directional Message Passing Neural Network (DimeNet)[54], Polarizable Atom Interaction Neural Network (PaiNN)[55], Geometric Message Passing Neural Network (GemNet)[56], SpinConv[57], Spherical Channel Network (SCN)[58], Neural Equivariant Interatomic Potentials (NequIP)[59], MACE[60] and Equiformer/EquiformerV2[61,62]. These networks are based on message passing among node and edge equivariant representations and have demonstrated promising fitting accuracy. However, it has been noted that GNNs are not easily parallelizable, making them less ideal for large-scale MD simulations[63].

Pre-training, or representation learning[64,65], has shown significant success across various applications, including natural language processing[30,66] and computer vision[67]. In the realm of molecular modeling, a primary objective of pre-trained models is to learn atomic representations of chemical species and 3D configurations of atoms.

One category of downstream tasks involves property prediction. Pre-trained models can be trained in an unsupervised manner by recovering masked atomic types and perturbed coordinates[68–72], by undertaking generative tasks[69], or by engaging in supervised learning tasks such as regression and classification[31,73–75].

Another category of downstream tasks focuses on the modeling of PESs. The model can be pre-trained through unsupervised tasks like denoising or chemical species restoration[25,28], supervised learning of energy, force, or partial charge[27,76], or a combination of both types of tasks[26]. Interestingly, most of these methods were developed for pre-training on molecule-in-vacuum systems, thus limiting the downstream tasks to such a class of tasks. Gardner et al.[76] developed pre-trained models for condensed-phase carbon systems, but these models are unlikely to be generalizable to systems composed of chemical elements other than carbon. Zhang et al.[43] pre-trained the DPA-1 model on the OC2M dataset[77] and examined its performance on downstream tasks involving high entropy alloys and AlMgCu ternary alloys. However, the study did not investigate downstream tasks related to non-metallic systems.

Results

The workflow of LAM

The LAM workflow includes the phases of *pre-training*, *fine-tuning* for downstream tasks, and *knowledge distillation*, as schematically presented in Fig. 1. The LAM is constructed with a unified descriptor that encodes the symmetry-preserving representation of the chemical and configurational spaces of atomic systems. This descriptor is connected to the energy-fitting networks, each predicting the energy (E) and force (F) outputs based on the data used during the pre-training phase (see Fig. 1a).

The LAM employs a multi-task training strategy, as illustrated in Fig. 1a. Specifically, the network parameters within the unified descriptor are concurrently optimized through back-propagation using all pre-training datasets. In contrast, the parameters of the fitting network are updated exclusively with the specific pre-training dataset to which they are

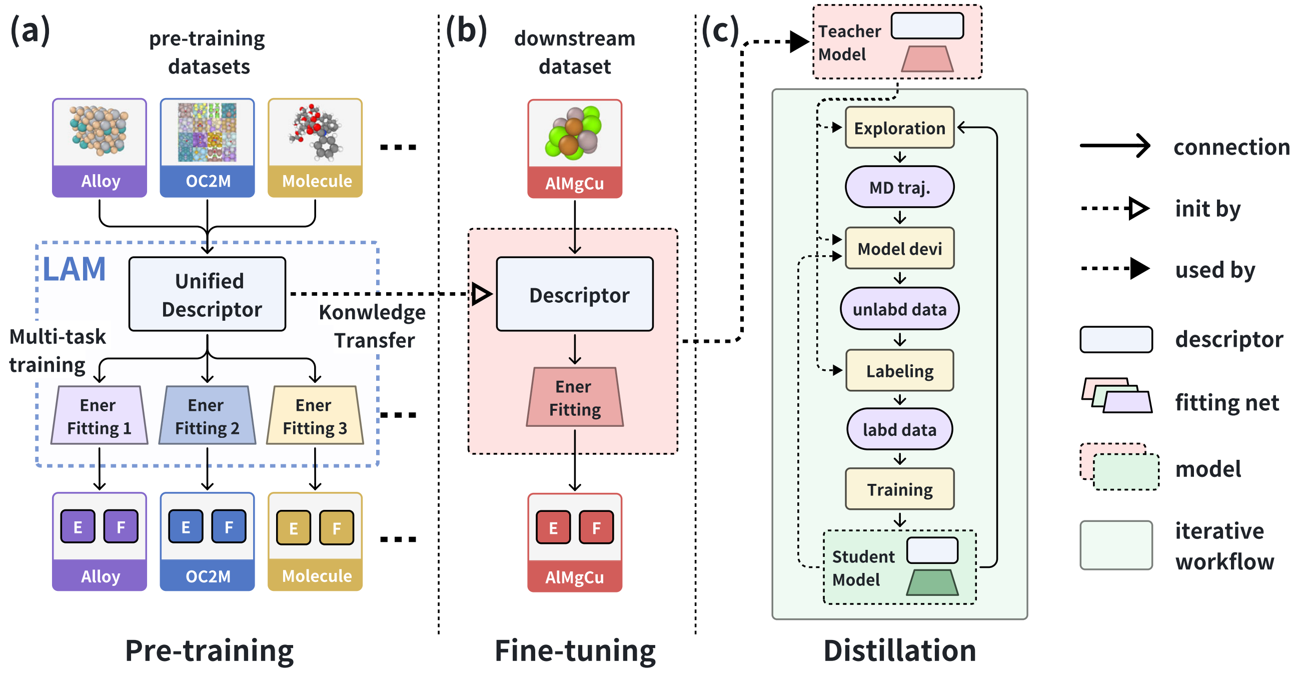

Fig. 1 | An overview of the proposed LAM workflow. a The multi-task pre-training process, in which different DFT-labeled data can be pre-trained together by sharing a single descriptor and having their unique fitting nets, with sampling according to their importance. This results in a unified descriptor. In this work, we have proposed the DPA-2 architecture for the unified descriptor. **b** The fine-tuning process on the downstream dataset, using the pre-trained unified descriptor and selecting a fitting net from upstream tasks or reinitializing the fitting net for the downstream dataset. **c** The distillation process uses the fine-tuned model as a teacher model, iteratively performing MD simulations and adding labeled data to the training set to train a high-efficiency student model, which is convenient for downstream applications.

associated. This approach is fundamentally different from the single-task training paradigm, where all model parameters, encompassing those within both the descriptor and the fitting network, are refined using a singular training dataset. The inability to merge the pre-training datasets into a unified "super-dataset" stems from the fact that labels across different datasets are typically derived from DFT calculations subject to variable conditions, such as exchange-correlation functionals, basis sets, and energy cut-off radii, culminating in distinct PESs. We have shown that the multi-task training is as efficient as the single-task training scheme, see section S3 of the Supplementary Materials. Therefore, multi-task training delivers the possibility of training the atomic representation from the heterogeneously labeled pre-training datasets. It is noted that although a hybrid multi-task pre-training approach using both labeled and unlabeled data is technically feasible, we focus on supervised learning for pre-training in this work, and leave the investigation of hybrid multi-task pre-training in future studies.

The pre-trained descriptor and the fitting networks can be fine-tuned for specific downstream PES modeling tasks, as illustrated in Fig. 1b. In the downstream model, the descriptor is initialized with the pre-trained unified descriptor, while the fitting network may be initialized either randomly or with a fitting head akin to the one used in one of the pre-training tasks. Given that the pre-training dataset encodes the bulk of the information within the descriptor, the initialization method for the downstream fitting network is likely to be of minor importance. The training dataset for a downstream task might be pre-existing and ready for training, or it could be generated through concurrent learning schemes such as DP-GEN[14]. In this study, we present several ready-to-use downstream datasets to validate the effectiveness of our proposed methodology and defer the exploration of concurrent learning-based data generation to future research.

The fine-tuned model, while possessing a large number of parameters, may exhibit reduced efficiency when directly applied to applications like MD simulations. To address this concern, we propose model distillation to create a streamlined version that retains the desired accuracy for downstream tasks while also enhancing processing speed and facilitating extensive simulations. Figure 1c depicts the distillation procedure, which employs an iterative learning loop. Within this framework, the original model, henceforth referred to as the "teacher", labels the data. In parallel, a "student" model, characterized by a simplified architecture (e.g., DPA-1 without any attention layer, which can be further compressed[78] to significantly enhance performance), is trained on this labeled data. The teacher model is then engaged in MD exploration, operating under conditions akin to those of the intended downstream application. This ensures that the chemical and physical parameters encountered during both the distillation process and the actual tasks are consistent, facilitating effective learning by the student model. Configurations from the MD trajectories are sampled, and the student model's predictions are compared against those of the teacher. If the discrepancy between their predictions surpasses a pre-established threshold, these configurations are appended to the training set for subsequent iterations. The cycle is reiterated until the student model's predictive accuracy either meets the preset standards or stabilizes without further improvement.

Datasets and DPA-2 descriptor

The primary goal in developing LAMs is to embed comprehensive knowledge within the multi-task pre-trained model by leveraging the pre-training dataset. Consequently, this embedded knowledge is anticipated to alleviate the intensive fine-tuning process required for specific downstream tasks. This objective necessitates two essential criteria during the pre-training phase: (1) on the data side, the pre-training dataset must encompass a broad spectrum of chemical and configurational spaces to represent potential scenarios in downstream applications; and (2) on the model side, DPA-2 model pre-trained in a multi-task manner, is expected to be robust and to exhibit a strong ability to generalize to downstream tasks, provided that its training data meet criterion (1).

For the first criterion, the datasets utilized in this study are summarized in Table 1. Detailed descriptions are provided in section S1 of the Supplementary Materials. Some datasets are newly generated in this work, including metallic alloys (Alloy), cathode materials (Cathode), metal nanoclusters (Cluster), and drug-like molecules (Drug). Some datasets are contributed by the DeepModeling community (https://github.com/

Table 1 | Overview of pre-training and downstream datasets employed in the multi-task learning framework

Pre-training datasets					
Name	**Element**	**#Train**	**#Test**	**#Total**	**Weight**
Alloy	53	71,482	1240	72,722	2.0
Cathode-P	Li, Na, O, Mn, Fe, Co, Cr, Ni	58,690	6451	65,141	1.0
Cluster-P	Pd, Ru, Al, Au, Ag, Pt, Si, Cu, Ni	139,200	14,936	154,136	1.0
Drug	H, C, N, O, F, Cl, S, P	1,379,956	24,257	1,404,213	2.0
FerroEle-P	15	6966	760	7726	1.0
OC2M	56	2,000,000	999,866	2,999,866	2.0
SSE-PBE-P	Li, P, S, Si, Ge	15,019	755	15,774	1.0
SemiCond-P	14	136,867	14,848	151,715	1.0
H2O-PD	H, O	46,077	2342	48,419	1.0
Ag ∪ Au-PBE	Ag, Au	16,696	812	17,508	0.2
Al ∪ Mg ∪ Cu	Al, Mg, Cu	24,252	1145	25,397	0.3
Cu	Cu	14,596	770	15,366	0.1
Sn	Sn	6449	276	6725	0.1
Ti	Ti	10,054	474	10,528	0.1
V	V	14,935	738	15,673	0.1
W	W	42,297	2100	44,397	0.1
C12H26	H, C	33,898	1598	35,496	0.1
HfO2	O, Hf	27,660	917	28,577	0.1
sum	73	4,045,094	1,074,285	5,119,379	13.2
Downstream datasets					
Name	**Element**	**#Train**	**#Test**	**#Total**	**Weight**
Cathode-D	Li, Na, O, Mn, Fe, Co, Cr	30,002	3244	33,246	1.0
Cluster-D	Pd, Au, Ag, Pt, Cu, Ni	4218	395	4613	1.0
FerroEle-D	15	7521	597	8118	1.0
SSE-PBE-D	Li, P, S, Sn	2563	131	2694	0.5
SSE-PBESol	Li, P, S, Si, Ge, Sn	7502	384	7886	0.5
SemiCond-D	P, N, Al, Te, In, Se, Sb, B, As	78,614	8495	87,109	1.0
ANI-1x	H, C, N, O	4,872,049	83,956	4,956,005	1.0
Transition-1x	H, C, N, O	7,632,328	967,454	8,599,782	1.0
H2O-DPLR	H, O	557	46	603	0.5
H2O-SCAN0	H, O	7002	347	7349	0.5
H2O-PBE0TS	H, O	133,000	7000	140,000	0.5
H2O-PBE0TS-MD	H, O	38,000	2000	40,000	0.5
AgAu-PBED3	Ag, Au	64,239	2256	66,495	0.3
AlMgCu-D	Al, Mg, Cu	113,942	2820	116,762	0.2
In2Se3	In, Se	11,621	568	12,189	0.2
Sum	39	13,003,158	1,079,693	14,082,851	9.0

The columns provide dataset name, coverage of the chemical space, number of training data points, number of test data points, the total data count, and assigned weight.

deepmodeling/AIS-Square/tree/main/datasets), including the ferroelectric perovskite (FerroEle), solid-state-electrolyte (SSE), semiconductors (SemiCond), H$_2$O, metallic material datasets (e.g., Sn, AgAu and AlMgCu), and the pyrolysis of n-dodecane (C12H26). Additionally, we have the open catalyst 20[77] (OC2M) that is formed by AIMD trajectories of molecular chemical reactions catalyzed by metallic substrates. These datasets are labeled with various DFT software like the VASP[79,80], Gaussian[81], and ABACUS[82,83]. In addition, They are divided into two groups, the pre-training and the downstream datasets, as detailed in section S1 of the Supplementary Materials. It is noted that the division is only to demonstrate the effectiveness of the workflow of LAM. For production purposes, all the datasets listed in Table 1 should be used to pre-train an LAM.

In the last column of Table 1, weights are assigned to each pre-training dataset. These weights are based on relevance, diversity in both chemical and configurational spaces, and data volume. The weight of a dataset is proportional to its selection probability during multi-task training, meaning that datasets with higher weights are favored in each training iteration. These weights also play a crucial role in calculating the weighted average of errors across all datasets, as shown in Table 2 and Tables S2 and S3 of the Supplementary Materials, which helps to provide an assessment of the model's overall accuracy.

For the second criterion, we propose the DPA-2 model with full details of the model architecture explained in "Methods" section. The descriptor of the model, which is supposed to encode the representation of the chemical and

Table 2 | Comparison of the zero-shot generalization errors on downstream tasks

Downstream	Pre-train (only for ST)	Energy RMSE [meV/atom] ↓					Force RMSE [meV/Å] ↓				
		Data std.	MACE (MPtrj)	DPA-2 (MPtrj)	DPA-2 ST	DPA-2 MT	Data std.	MACE (MPtrj)	DPA-2 (MPtrj)	DPA-2 ST	DPA-2 MT
WARMSE		**121.4**	**104.0**	**68.3**	**100.2**	**50.1**	**1405.4**	**575.6**	**516.6**	**628.0**	**238.8**
AgAu-PBED3	AgAu-PBE	906.9	1812.8	268.9	222.9	192.3	878.0	683.2	293.3	236.9	63.6
AlMgCu-D	AlMgCu	383.8	33.8	32.0	254.3	41.2	1229.5	240.1	245.3	663.7	111.8
AlMgCu-D	Alloy	383.8	33.8	32.0	74.9	48.4	1229.5	240.1	245.3	122.3	112.8
ANI-1x	Drug	198.9	52.3	61.7	67.2	56.6	2124.6	636.1	700.1	738.7	346.7
Cathode-D	Cathode-P	42.2	15.8	29.7	39.8	43.8	641.9	288.4	613.9	339.7	273.9
Cluster-D	Cluster-P	636.0	323.7	262.7	41.4	40.5	3605.4	2230.8	1193.6	238.4	190.5
FerroEle-D	FerroEle-P	43.0	12.5	14.5	6.3	3.9	881.3	191.3	194.2	282.7	115.1
H2O-DPLR	H2O-PD	15.6	2.1	2.0	9.1	9.3	825.2	94.4	99.7	263.5	263.4
H2O-H2O	H2O-PD	47.0	4.9	7.2	4.9	4.7	1941.0	381.0	382.7	58.8	64.4
H2O-PBE0TS-MD	H2O-PD	3.3	1.1	1.5	0.5	0.6	816.1	330.8	314.4	37.6	40.8
H2O-SCAN0	H2O-PD	12.6	3.2	3.8	1.1	0.7	2163.2	387.5	385.2	409.2	162.9
In2Se3	SemiCond-P	120.5	31.9	24.5	160.6	38.9	611.1	190.2	188.0	1544.1	341.6
SemiCond-D	SemiCond-P	587.6	49.8	70.9	486.2	175.7	1755.4	470.7	534.9	1439.4	439.3
SSE-PBE-D	SSE-PBE-P	79.0	33.7	39.4	40.7	6.2	789.5	222.1	249.9	635.6	162.4
SSE-PBESol	SSE-PBE-P	84.3	32.5	37.4	26.1	8.3	810.9	231.8	260.4	425.0	115.3
Transition-1x	Drug	139.8	56.4	55.1	48.2	45.8	368.1	518.6	618.3	1298.6	363.8

The MACE-MP-0 (MACE) and DPA-2 pre-trained on the MPtrj dataset, and the DPA-2 pre-trained by single-task (ST) and multi-task (MT) approaches are compared. The DPA-2 ST is trained by the pre-training datasets listed in the second column of the Table, while the DPA-2 MT is trained by all the pre-training datasets listed in Table 1. The energy and force RMSEs on the downstream test datasets are reported. The weighted averaged RMSEs (WARMSE) with the weights presented in Table 1 are given in the first row of the table. The standard deviations of energy and force labels in the test set are also provided. If the RMSE is smaller than the corresponding standard deviation, the model shows the ability of zero-shot generalization, on the other hand, the model cannot be generalized to downstream tasks without downstream data.

configurational spaces of the pre-training dataset, is schematically demonstrated in Fig. 2. The chemical and configurational spaces are represented by a single-atom channel f_i, a rotationally invariant pair-atom channel g_{ij} and a rotationally equivariant pair-atom channel h_{ij}. The pair-atom representations are initialized by the environment matrix (operator env in Fig. 2), which encodes the relative positions of the near neighbors within a certain cut-off radius (r_c^0 and r_c^1), and smoothly decays to zero at the cut-off radius. The single-atom representations f_i is initialized by a repinit (representation initializer) layer. Then the single- and pair-atom representations are subsequently updated by the representation transformer (repformer) layers, which are stacked 12 times and communicate information in a message-passing manner between the layers. In each of the repformer layers, f_i is updated by convolution, symmetrization, MLP, and localized self-attention operators, while g_{ij} is updated by MLP, dot-product, and gated self-attention operators (see Fig. 2c and "The architecture of the DPA-2 model" section for more details). The contribution of different building blocks to the model accuracy is investigated by an ablation study in section S8 of the Supplementary Materials.

The DPA-2 model is designed to be extensible and inherently respects translational, rotational, and permutational symmetries. Moreover, it is conservative, as it predicts atomic forces by computing the negative gradient of the system's energy with respect to the atomic positions, $F_i = -\nabla_{r_i} E$, and calculates the virial tensor $\Xi_{\alpha\beta} = \sum_\gamma (-\nabla_{h_{\gamma\alpha}} E) h_{\gamma\beta}$, where E represents the energy, r_i denotes the position of atom i, and $h_{\alpha\beta}$ is the βth component of the αth basis vector of the simulation cell. Furthermore, all components of the DPA-2 model are continuous up to the second-order derivative, ensuring energy conservation. Numerical examples demonstrating the energy conservation properties of the DPA-2 model can be found in the Supplementary Material section S9.

Generalizability of the multi-task pre-trained DPA-2 model

Before moving to a discussion on the generalizability of the multi-task training scheme, we test the model of DPA-2 by using single-task benchmarks, which are directly comparable to the state-of-the-art model architectures. In the first benchmark, the ANI-1x dataset, the DPA-2 shows superior test accuracy compared with the ANI-1x model reported in Ref. 11, see Table S1 in the Supplementary Materials. In the second benchmark, the accuracy of the DPA-2 model is comparable to GemNet-OC[84] and higher than Equiformer V2[62], NequIP[59], Allegro[63], and MACE[60] models on the pre-training datasets, see Table S2 in the Supplementary Materials.

Next, we train the DPA-2 model on all the pre-training datasets by the multi-task scheme. The details of the training protocol, the test accuracy of these datasets, and a discussion on the effectiveness of the multi-task scheme are given in section S3 of the Supplementary Materials. The multi-task training introduces minimal degradation in performance on the pre-training datasets compared to the single-task training. As reported in Table S3, it achieves nearly the same force accuracy as single-task training, with a WARMSE of 116.3 meV/Å compared to 111.1 meV/Å for single-task training, while it is 25% less accurate in terms of energy prediction, with a WARMSE of 18.6 meV/atom vs 14.9 meV/atom in single-task training. The representation learned by the multi-task trained DPA-2 model is visualized and analyzed in the Supplementary Materials section S4.

We investigate the generalizability of the multi-task pre-trained DPA-2 model to downstream tasks by testing the model directly on downstream datasets. This approach is known as zero-shot generalization because no data from the downstream tasks are used to refine the pre-trained model before testing. In an ideal scenario, a perfectly generalizable model—that is, one that encapsulates the chemical knowledge of the periodic table and all relevant configurations for a given downstream task—would exhibit a zero-shot generalization error comparable to, or potentially lower than, the test error of a model specifically trained from scratch for that task.

The zero-shot generalizability of the multi-task pre-trained DPA-2 model is presented in Table 2 and compared with its single-task pre-trained counterpart, MPtrj-trained DPA-2, and MACE-MP-0. Further comparisons with M3GNet and CHGNet are provided in the Supplementary Materials Table S4. For all cases, the single-task DPA-2 models are exclusively trained on the datasets specified in the second column, whereas the multi-task DPA-2 model undergoes pre-training on the entire corpus of pre-training datasets (see Table 1). The multi-task DPA-2 model then

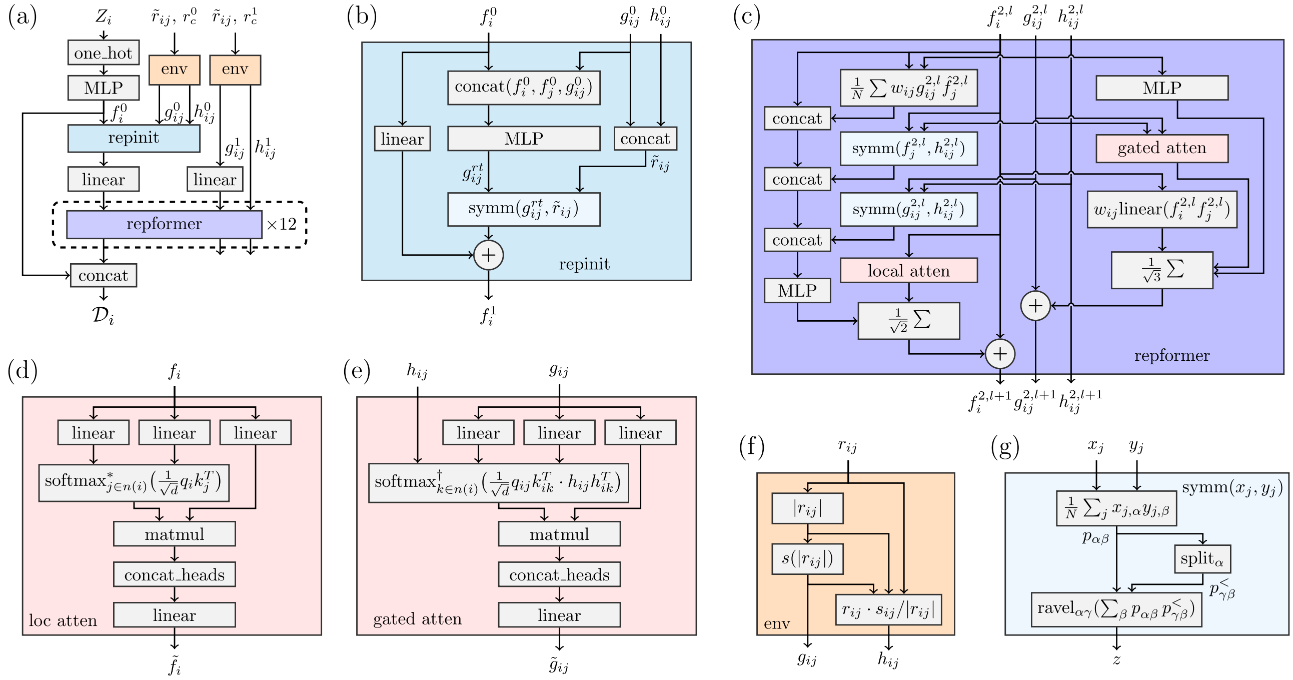

Fig. 2 | Architecture and components of the DPA-2 descriptor. a Detailed architecture of the DPA-2 descriptor, which includes two primary components: repinit and repformer. **b** Structure of repinit. **c** Structure of repformer. **d**–**g** Substructures referenced in subsequent sections.

employs the fitting head indicated in the second column to initialize the fitting procedure for downstream tasks. All model variants are evaluated on their respective downstream datasets without any additional training. The results demonstrate that multi-task training substantially enhances generalizability compared to the single-task pre-trained DPA-2 and the MPtrj-trained models. The comparable performance between the MPtrj-trained MACE-MP-0 and DPA-2 suggests that the improvement is primarily due to the multi-task pre-training scheme rather than differences in model architecture.

Fine-tuning downstream tasks

Although zero-shot generalizability is often observed to a certain extent, a gap from perfect generalization typically remains. To bridge this gap, we fine-tune the models using data from the downstream tasks. A stronger generalizability in a pre-trained model implies that less data is required during fine-tuning, leading to higher sample efficiency. The reduction in sample size relative to training a model from scratch quantifies the advantage of employing a multi-task pre-trained model.

The sampling efficiency of the pre-trained DPA-2 on downstream tasks was evaluated by comparing it against various other DP models that were trained from scratch. Figure 3 showcases a selection of downstream tasks, with a comprehensive comparison available in section S5 of the Supplementary Materials. The figure illustrates the convergence trends of the energy and force RMSEs in relation to the expanding sample size used for downstream training.

To draw distinctions between the fine-tuned DPA-2 and the from-scratch DPA-2 models, it is important to realize that both models share identical architectures. However, the fine-tuned model begins with parameters derived from a multi-task pre-trained model, whereas the from-scratch model starts with randomly initialized parameters. The fine-tuned DPA-2 model consistently achieves lower error curves compared to the DPA-2 model trained from scratch, particularly when the available downstream data is scarce. This translates to a considerable reduction in the amount of data needed to reach equivalent levels of accuracy. Taking the H2O-PBE0TS-MD task for example, two orders of magnitudes of training data are saved to reach the same energy accuracy, see the zoomed-in Fig. 3. As the sample size grows, the performance disparity between the fine-tuned and from-scratch DPA-2 models diminishes. This outcome is anticipated, given that both models possess the same capacity and, theoretically, their accuracy should converge as the dataset approaches an infinite size. When comparing DeepPot-SE (DP-SE), DPA-1, and DPA-2 models trained from scratch, the DPA-2 model exhibits superior performance over the other architectures. While the convergence patterns of the DPA-1 and DP-SE models are somewhat parallel, the DP-SE model reaches a performance plateau more rapidly than the DPA-1 in the FerroEle-D, SSE-PBESol, and SemiCond-D tasks.

Model distillation and evaluation

The fine-tuned DPA-2 model typically suffers from computational inefficiency due to its extensive parameter set, as illustrated in Fig. 4f. To address this, we employed a knowledge distillation approach, transferring insights from the fine-tuned DPA-2 models to compressed DPA-1 models without attention layers. We evaluated the performance of these distilled models in terms of efficiency and accuracy on three benchmark downstream tasks: H2O-PBE0TS-MD, SSE-PBE-D, and FerroEle-D. Notably, in all the cases, the fine-tuned models are exposed to only a small portion (0.25% to −7.86%, see Table S5) of the downstream dataset, and are used to generate the distillation training datasets that sufficiently cover the relevant configuration spaces. In the FerroEle-D task, we append the full FerroEle-P to a small (7.86%) portion of the FerroEle-D dataset for the training of the fine-tuned model. The FerroEle-D that contains solid solution perovskite oxides was generated by the concurrent learning scheme starting from the FerroEle-P dataset that contains unitary perovskite (see ref. 85 and Supplementary Materials section S1). Consequently, the FerroEle-D dataset alone does not provide a comprehensive basis for training a fully capable potential model. Additional details on the model distillation process are provided in Supplementary Materials S7.

After distillation, the time-to-solution and the maximal system size that can be simulated on a single GPU card improved by nearly two orders of magnitude, as shown in Fig. 4f. Moreover, the accuracy of the distilled models is on par with that of the fine-tuned DPA-2 models, as detailed in Table S5. The distilled models appear to have reached the peak of their performance, given that their accuracies closely match those of the DPA-1

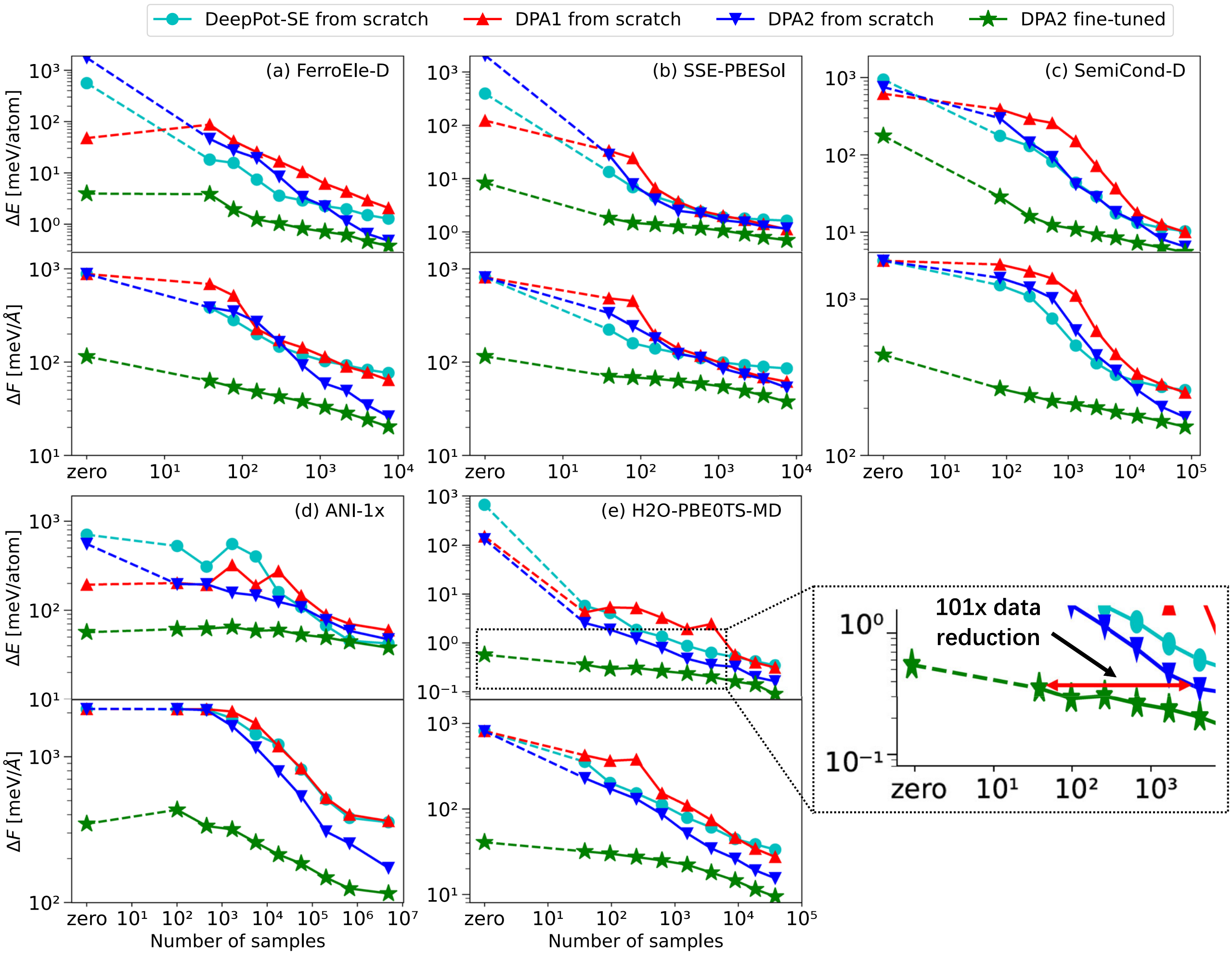

Fig. 3 | Comparative analysis of sample efficiency on downstream tasks. The horizontal axis represents the volume of downstream data required, while the vertical axis depicts the RMSE in energy or force predictions. For a uniform assessment across models, the number of training epochs per model for each downstream task is normalized to a standard value, derived by dividing 1 million by the number of downstream samples.

models (without an attention layer) when trained on the complete downstream datasets.

Finally, to validate the reliability of the distilled models beyond the energy and force RMSEs, we have conducted various application tests on the aforementioned three systems, as reported in Fig. 4a–e. In the downstream task of H2O-PBE0TS-MD, we observe that the radial distribution functions (RDFs) and the angular distribution function (ADF) of the distilled model are in almost perfect agreement with those obtained from the AIMD simulation, see Fig. 4a, b. In the downstream task of SSE-PBE-D, the diffusion constants of Lithium ions in the $Li_{10}SnP_2S_{12}$ system under different temperature conditions are calculated. The distilled model presents satisfactory agreement with the previously reported MD simulations using the DP-PBE LiSnPS model and DFT (i.e., AIMD simulations)[86,87], see Fig. 4c. The discrepancy between the simulation and the experimental results[88] may be attributed to the approximation error of the density functional and finite size effects, as discussed in ref. 89. In the downstream task of FerroEle-D, we investigated the temperature-driven phase transition in the solid solution ferroelectric perovskite $Pb(In_{1/2}\ Nb_{1/2})O_3$–$Pb(Mg_{1/3}\ Nb_{2/3})O_3$–$PbTiO_3$ (PIN–PMN–PT), see Fig. 4(d–e). Tetragonal-cubic (T-C) transitions are observed at ~250 K and ~300 K for two concentrations 0.29PIN–0.45PMN–0.26PT and 0.36PIN–0.36PMN–0.28PT, respectively. The fact that the transition temperature rises for ~50 K due to the increment in the PIN ($Pb(In_{1/2}\,Nb_{1/2})O_3$) portion from 29% to 36% is in line with the experimental observations[90,91].

Discussion

In this work, we introduce DPA-2, a newly designed model architecture for the Large Atomic Model (LAM), supported by a comprehensive pipeline that includes multi-task pre-training, fine-tuning, knowledge distillation, and practical deployment. The principal findings concerning DPA-2 are as follows: (1) DPA-2 demonstrates exceptional ability for generalization, primarily due to the multi-task pre-training approach, which utilizes 18 datasets covering 73 chemical elements. These datasets would not typically be merged in a single-task pre-training scenario due to differing labeling methodologies, such as exchange-correlation functionals, energy cutoffs, and k-space grid spacing. (2) The multi-task pre-training approach significantly enhances zero-shot generalization on downstream tasks. It reduces the zero-shot weighted average RMSEs by 52% in energy and 59% in force compared to the MPtrj pre-trained MACE model, and by 50% in energy and 62% in force compared to the single-task pre-trained DPA-2 model. (3) In downstream tasks, the multi-task pre-training approach enables a reduction in data requirements by approximately 1–2 orders of magnitude without sacrificing accuracy. These results suggest that the DPA-2 model, along with the proposed workflow, stands as a promising framework for molecular and materials simulation.

It is evident that the existing pre-training datasets for the DPA-2 model are insufficient. For example, the datasets currently in use are notably deficient in information on 2-D materials, which significantly limits the model's generalizability to such systems. As a result, the development of LAMs like

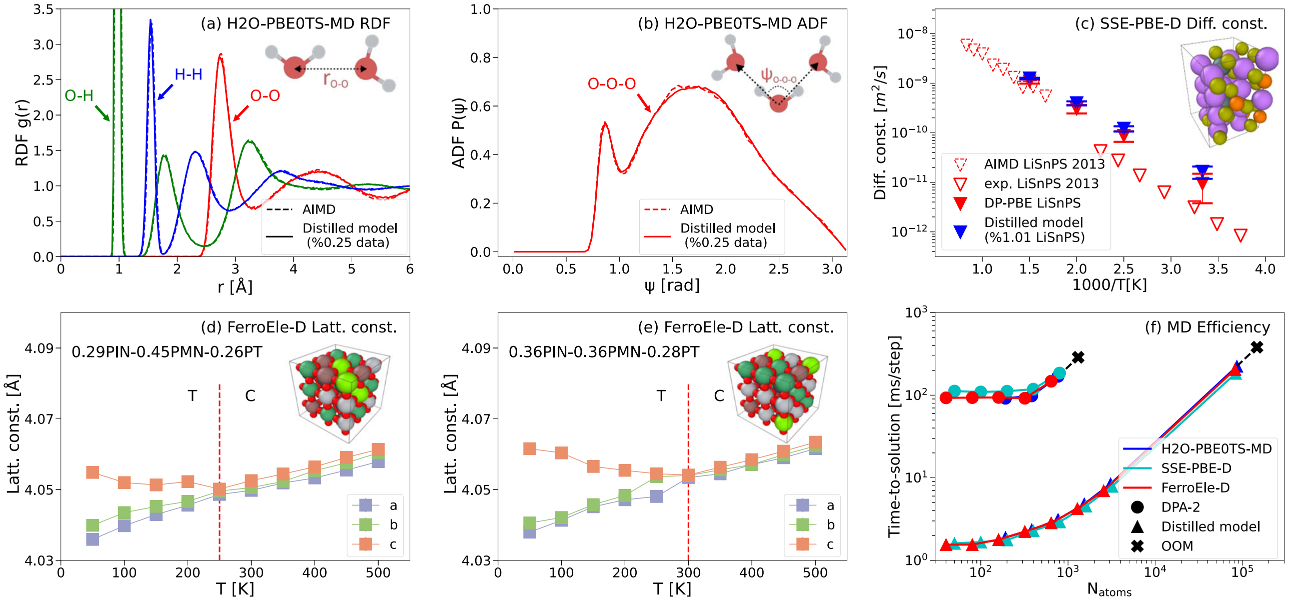

Fig. 4 | Evaluation of the distilled model across various downstream applications. **a**, **b** Comparison of the radial distribution function (RDF) and angular distribution function (ADF) for the H2O-PBE0TS-MD dataset between the reference AIMD results[93] and the distilled model. The model is distilled from a DPA-2 model fine-tuned from merely 0.25% of DFT-labeled data. **c** A comparison of diffusion constants for the solid-state electrolyte $Li_{10}SnP_2S_{12}$. The constants were determined using various methods: the distilled model, DPMD as reported in Huang et al.[89], AIMD simulations from the studies by Mo et al. and Marcolongo et al.[86,87], and experimental findings from solid-state nuclear magnetic resonance (NMR) as documented by Kuhn et al.[88]. The distilled model is trained from a DPA-2 model fine-tuned by 1.01% of the SSE-PBE-D data. **d**, **e** The temperature-dependent lattice constants for the ternary solid solution ferroelectric perovskite oxides $Pb(In_{1/2}\ Nb_{1/2})O_3$–$Pb(Mg_{1/3}\ Nb_{2/3})O_3$–$PbTiO_3$ (PIN–PMN–PT). The NPT MD simulations using the distilled model are conducted for two concentrations, 0.29PIN–0.45PMN–0.26PT and 0.36PIN–0.36PMN–0.28PT[85]. The model is distilled from a DPA-2 model fine-tuned with the complete FerrEle-P dataset and 7.86% of the FerrEle-D data. **f** Computational efficiency assessment for the aforementioned three systems, showcasing the time-to-solution as a function of the system size in the number of atoms (N_{atoms}).

DPA-2 must be considered a long-term endeavor. This process necessitates the ongoing collection of diverse training data, the incorporation of application-specific test cases, and the establishment of automated workflows for data preprocessing, model training, model evaluation, and version updates. In recognition of these needs, we underscore the importance of fostering LAMs within an open and collaborative ecosystem. Such an approach would enable the molecular simulation community to both benefit from and contribute to the evolution of LAMs. Reflecting our commitment to this vision, we have launched the OpenLAM Initiative (https://deepmodeling.github.io/blog/openlam). Updates on this initiative will be regularly posted on the AIS Square platform (https://www.aissquare.com/openlam). We cordially invite readers to participate in this project in any capacity they deem fit.

Methods

Formulation

In this study, we examine a system consisting of N atoms, where the atomic numbers are represented by the list $\mathcal{Z} = (Z_1, \ldots, Z_i, \ldots, Z_N)$, and the atomic coordinates are denoted by the list $\mathcal{R} = (r_1, \ldots, r_i, \ldots, r_N)$. The potential energy surface (PES) of the system is symbolized by E, a function dependent on elemental types and coordinates, expressed as $E = E(\mathcal{X})$, $\mathcal{X} := (\mathcal{R}, \mathcal{Z})$. The potential energy surface can be further decomposed into the following equation:

$$E = \sum_i E_i, \tag{1}$$

where E_i signifies the atomic energy contributions originating from atom i. The atomic force exerted on atom i, represented as F_i, is defined as the negative gradient of the total energy with respect to the coordinate:

$$F_i = -\nabla_{r_i} E. \tag{2}$$

For periodic systems, the virial tensor can be obtained as follows:

$$_{\alpha\beta} \quad = -\sum_{\gamma} \frac{\partial E}{\partial h_{\gamma\alpha}} h_{\gamma\beta}, \tag{3}$$

where $\Xi_{\alpha\beta}$ corresponds to the $\alpha\beta$ component of the virial tensor, and $h_{\alpha\beta}$ yields the β-th component of the α-th cell vector.

The architecture of the DPA-2 model

The DPA-2 is a model that predicts the atomic energy contribution based on the atomic numbers $\mathcal{Z}$ and the coordinates $\mathcal{R}$. It consists of two parts,

$$E_i = \mathcal{F}\left(\mathcal{D}_i(\mathcal{R}, \mathcal{Z})\right), \tag{4}$$

where $\mathcal{D}_i$ represents the descriptor of atom i. The descriptor must be a smooth mapping from the atomic numbers and coordinates to a hidden representation that remains invariant under translational, rotational, and permutational (only among atoms with the same atomic number) operations.

The fitting network $\mathcal{F}$ is usually modeled by a standard multiple-layer perceptron (MLP) composed of an energy-biasing layer,

$$\mathcal{F}(\mathcal{D}_i) = e_{\text{bias}}\left(\text{MLP}(\mathcal{D}_i)\right). \tag{5}$$

The energy bias layer "e_{bias}" adds a constant bias to the atomic energy contribution according to the atomic number, i.e., $e_{\text{bias}}(Z_i)(\text{MLP}(\mathcal{D}_i)) = \text{MLP}(\mathcal{D}_i) + e_{\text{bias}}(Z_i)$. Ideally, the energy bias e_{bias} should be taken as the energy of an atom in a vacuum. In practice, the energy bias may be determined by a least-square fitting of the energies in the training data. More precisely, suppose we have M data frames, and within the m-th frame, we have c_{mz} atoms with atom number z, and the DFT labeled energy of the frame is denoted by E_m^*. Then the linear system

$$\sum_z c_{mz} e_{\text{bias}}(z) = E_m^*, \quad m = 1, \ldots, M, \tag{6}$$

is solved in the least-square sense. Here we assume that the number of independent equations in system Eq. (6) is equal to or smaller than the number of frames M.

The DPA-2 descriptor is graphically illustrated in Fig. 2, specifically,

$$\mathcal{D}_i = \text{concat}\left(f_i^0, f_i^2\right), \tag{7}$$

where f_i^0 and f_i^2 denote the single-atom representations of atom i. The requirements for smoothness and symmetry preservation in single-atom representations are identical to those for the descriptor. The representation f_i^0 is defined as

$$f_i^0 = \text{MLP}\left(\text{one\_hot}(Z_i)\right). \tag{8}$$

The atomic number, Z_i, is initially converted into a one-hot representation and subsequently embedded by an MLP. The output f_i^0 is the single-atom hidden representation with dimension n_1^0. The single-atom representation is updated by the repinit (representation-initializer) layer that encodes the information of local configuration, expressed by the pair-atom representations g_{ij}^0 a and h_{ij}^0, into the single-atom representation.

$$f_i^1 = \text{repinit}\left(f_i^0, g_{ij}^0, h_{ij}^0\right). \tag{9}$$

The feature f_i^2 is mapped from single-atom representation and pair-atom representations g_{ij}^0, h_{ij} by a multiple-layer structure,

$$f_i^2 = \underbrace{\text{repformer} \circ \cdots \circ \text{repformer}}_{\times 12}\left(\text{linear}\left(f_i^1\right), \text{linear}\left(g_{ij}^1\right), h_{ij}^1\right), \tag{10}$$

where the single- and pair-atom representations are updated by repformer (representation-transformer) layers. The repformer is designed in a way that the input and output representations share the shape dimension, thus they are stacked 12 times. The "∘" in Eq. (10) thus denotes the layer composition (or mathematically the function composition). The linear mappings are used to change the dimension of f_i^1 and g_{ij}^1 to match the shape requirement of repformer. The pair-atom representations $g_{ij}^0, h_{ij}^0, g_{ij}^1$ and h_{ij}^1 will be introduced shortly later. It is assumed that the repinit and repformer layers only require the information of i's neighboring atoms, i.e., all atoms falling within a sphere centered at atom i with a radius r_c. This radius is commonly referred to as the cut-off radius. We thus introduce the notation $N_{r_c}(i)$, which represents the set of all neighbors of i, i.e., $N_{r_c}(i) = \{j : j \neq i, |r_j - r_i| < r_c\}$. The maximum possible number of neighbors for the atoms in the system is denoted by $N_{r_c}^m$, so we have $|N_{r_c}(i)| \leq N_{r_c}^m, \forall i$.

To define the pair-atom representations, g_{ij}^0, h_{ij}^0, we consider the local configuration of atom i represented by the augmented environment matrix with shape $N_{r_c^0}^m \times 4$, where r_c^0 is the cut-off radius used to compute the pair-atom representations. The j-th row of the environment matrix, being a 4-dimensional vector, is defined by

$$\tilde{r}_{ij} = s(r_{ij}) \times \left(1, \frac{x_{ij}}{|r_{ij}|}, \frac{y_{ij}}{|r_{ij}|}, \frac{z_{ij}}{|r_{ij}|}\right), \tag{11}$$

where (x_{ij}, y_{ij}, and z_{ij}) are the Cartesian coordinates of the relative position $r_{ij} = r_i - r_j$. In most cases, the number of neighbors is smaller than $N_{r_c}^m$, so the environment matrix only has $|N_{r_c}(i)|$ rows defined by Eq. (11), and the remaining positions are filled with zeros. The switched inverse distance function s in Eq. (11) is defined by

$$s\left(r_{ij}\right) = \frac{w_{ij}}{|r_{ij}|}, \quad w_{ij} = w(|r_{ij}|). \tag{12}$$

The switch function w takes the value 0 outside the cut-off radius r_c, and 1 inside a starting point of switching, denoted by r_{cs}. In between r_{cs} and r_c, the switch function smoothly changes from 1 to 0. It is required that w has a continuous second-order derivative $\mathbb{R}$. One possible implementation of w is provided as

$$w\left(|r_{ij}|\right) = \begin{cases} 1 & \text{if } r_{ij} < r_{cs}, \\ u^3\left(-6u^2 + 15u - 10\right) + 1 & \text{if } r_{cs} \leq r_{ij} < r_c, \\ 0 & \text{if } r_c \leq r_{ij}, \end{cases} \tag{13}$$

where $u = (|r_{ij}| - r_{cs})/(r_c - r_{cs})$ and $r_{cs} < r_c$ is the starting point of the smooth switch.

The first column of the augmented environment matrix is defined as the rotationally invariant pair-atom representation, while the remaining three columns are denoted by the rotationally equivariant pair-atom representation, i.e.,

$$g_{ij}^0 = s(r_{ij}), \tag{14}$$

$$h_{ij}^0 = s(r_{ij}) \times \left(\frac{x_{ij}}{|r_{ij}|}, \frac{y_{ij}}{|r_{ij}|}, \frac{z_{ij}}{|r_{ij}|}\right). \tag{15}$$

The procedure for calculating pair-atom representations is graphically illustrated in Fig. 2f. The representations g_{ij}^1 and h_{ij}^1 are established in precisely the same manner as g_{ij}^0 and h_{ij}^0, with the only potential variation being the selection of a distinct cut-off radius, denoted as r_c^1.

The repinit layer. This layer only updates the single-atom f_i^0 and pair-atom g_{ij}^0 representations, and does not update the equivariant pair-atom representation h_{ij} that is of dimension 3. The repinit layer first embeds the concatenated single- and pair-atom representations to update the pair-atom representation

$$g_{ij}^{rt} = \text{MLP}\left(\text{concat}\left(f_i^0, f_j^0, g_{ij}^0\right)\right), \quad \forall j \in N_{r_c^0}(i) \tag{16}$$

Then, we concatenate the g_{ij}^0 and h_{ij} pair-atom representations to recover the environment matrix and update the single-atom representation using a symmetrization operation

$$f_i^1 = \text{linear}\left(f_i^0\right) + \text{symm}\left(g_{ij}^{rt}, \tilde{r}_{ij}\right). \tag{17}$$

The symmetrization operator, first introduced by ref. 41, has the general form of symm (x_j, y_j), where x_j and y_j are neighbor-indexed vectors. It is assumed that x_j is rotationally invariant, while y_j is not, but the inner product is rotationally invariant. The symmetrization operator is defined by

$$\text{symm}(x_j, y_j) = \underset{\alpha\gamma}{\text{flatten}}\left(\sum_\beta p_{\alpha\beta} p_{\gamma\beta}^<\right), \tag{18}$$

$$p_{\alpha\beta} = \frac{1}{N_{r_c^0}^m} \sum_{j \in N_{r_c^0}(i)} w_{ij}\, x_{j,\alpha}\, y_{j,\beta}, \tag{19}$$

$$p^{<}_{\alpha\beta} = \underset{\alpha}{\mathrm{split}}(p_{\alpha\beta}). \tag{20}$$

In Eq. (18), the matrix dimensions α and γ are flattened to form a vector. In Eq. (19), the summation is taken over the index of neighbors j, making the matrix p permutationally invariant. When an atom comes into the neighborhood of atom i, the quantities x_j and y_j generally do not smoothly switch from 0. To prevent the discontinuous jump, the switch w_{ij} is multiplied. In Eq. (20), the matrix $p_{\alpha\beta}$ is split along the α dimension, and the first certain number of elements are taken and assigned with the notation $p^{<}$. It can be proven that the symmetrization operator is invariant with respect to rotational operations and permutational operations over atoms of the same atomic number[41].

The repformer layer. This type of layer maintains the input and output dimensions of the single- and pair-atom representations, allowing it to be stacked to enhance its representational capabilities. However, the output of the repinit may not necessarily satisfy the dimension requirements of the repformer layer. To address this issue, the representations are first projected to the desired shape using a linear layer, as follows:

$$f^{2,0}_i = \mathrm{linear}\left(f^1_i\right), \tag{21}$$

$$g^{2,0}_{ij} = \mathrm{linear}\left(g^1_{ij}\right), \tag{22}$$

$$h^{2,0}_{ij} = h^1_{ij}. \tag{23}$$

Subsequently, these representations are updated by the repformer layers. The dimensions of the single- and pair-atom representations are denoted by n^2_1 and n^2_2, respectively. In the subsequent discussion, the input representations for the l-th repformer layer are denoted by $f^{2,l}_i$ and $g^{2,l}_{ij}$.

In each repformer layer, the single-atom representation is updated by

$$f^{2,l+1}_i = \frac{1}{\sqrt{3}}\left(f^{2,l}_i + \mathrm{MLP}\left(\tilde{f}^{2,l}_i\right) + \mathrm{loc\_attn}\left(f^{2,l}_i\right)\right). \tag{24}$$

The intermediate representation $\tilde{f}^{2,l}_i$ is defined by

$$\tilde{f}^{2,l}_i = \mathrm{concat}\left(f^{2,l}_i, \frac{1}{N^m_{r^1_c}}\sum_{j\in N_{r^1_c}(i)} w_{ij} g^{2,l}_{ij}\hat{f}^{2,l}_j, \mathrm{symm}\left(f^{2,l}_j, h^{2,l}_{ij}\right), \mathrm{symm}\left(g^{2,l}_{ij}, h^{2,l}_{ij}\right)\right), \tag{25}$$

where $\hat{f}^{2,l}_j$ is a linearly transformed $f^{2,l}_j$ that has the same dimension as the equivariant pair-atom channel, i.e., $\hat{f}^{2,l}_j = \mathrm{linear}(f^{2,l}_j)$. The last term in Eq. (24) is the local multi-head self-attention, defined by

$$\mathrm{loc\_attn}\left(f^{2,l}_i\right) = \underset{\beta,h\to n^2_1}{\mathrm{linear}}\left(\sum_{j\in N_{r^1_c}(i),\alpha} B^{l,\eta}_{ij} f^l_{j,\alpha}\hat{V}^{l,\eta}_{\alpha,\beta}\right), \tag{26}$$

with the attention map B given by

$$\hat{q}^{l,\eta}_{i,\gamma} = \sum_\alpha f^l_{i,\alpha}\,\hat{Q}^{l,\eta}_{\alpha,\gamma}, \quad \hat{k}^{l,\eta}_{j,\gamma} = \sum_\beta f^l_{j,\beta}\,\hat{K}^{l,\eta}_{\beta,\gamma}, \tag{27}$$

$$B^{l,\eta}_{ij} = \underset{j\in N_{r^1_c}(i)}{\mathrm{softmax}^*}\left(\frac{1}{\sqrt{\hat{d}}}\sum_\gamma \hat{q}^{l,\eta}_{i,\gamma}\hat{k}^{l,\eta}_{j,\gamma}\right). \tag{28}$$

Here, $\hat{d}$ denotes the hidden dimension of the local self-attention, and the $\hat{Q}$, $\hat{K}$, and $\hat{V}$ are trainable matrices. The "*" over the softmax operator indicates that the softmax used in Eq. (28) is modified to guarantee the smoothness of the attention map. The definition will be introduced in Eq. (35) at the end of this subsection.

In each layer, the rotationally invariant pair-atom representation is updated by

$$g^{2,l+1}_{ij} = \frac{1}{\sqrt{4}}\left(g^{2,l}_{ij} + \mathrm{MLP}(g^{2,l}_{ij}) + w_{ij}\underset{n^2_1\to n^2_2}{\mathrm{linear}}(f^{2,l}_i \odot f^{2,l}_j) + \mathrm{gated\_attn}\left(g^{2,l}_{ij}, h_{ij}\right)\right), \tag{29}$$

where the last term in Eq. (29) is the gated multi-head self-attention, which is defined by

$$\mathrm{gated\_attn}\left(g^{2,l}_{ij}, h_{ij}\right) = \underset{\beta,h\to n^2_2}{\mathrm{linear}}\left(\sum_{k\in N_{r^1_c}(i),\alpha} A^h_{ijk} g^{2,l}_{ik,\alpha} V^{l,\eta}_{\alpha,\beta}\right). \tag{30}$$

In Eq. (30), the attention map A is given by

$$q^{l,\eta}_{ij,\gamma} = \sum_\alpha g^{2,l}_{ij,\alpha}\,Q^{l,\eta}_{\alpha,\gamma}, \quad k^{l,\eta}_{ik,\gamma} = \sum_\beta g^{2,l}_{ik,\beta}\,K^{l,\eta}_{\beta,\gamma}, \tag{31}$$

$$A^{l,\eta}_{ijk} = \underset{k\in N_{r^1_c}(i)}{\mathrm{softmax}^\dagger}\left(\left(\frac{1}{\sqrt{d}}\sum_\gamma q^{l,\eta}_{ij,\gamma}k^{l,\eta}_{ik,\gamma}\right)\left(\sum_\delta h_{ij,\delta}h_{ik,\delta}\right)\right), \tag{32}$$

where d denotes the hidden dimension of the self-attention, the Q, K, and V are trainable matrices, and η is the index of the attention heads. The gate term $h_{ij}h^T_{ik}$ is proved to be critical to the generalization ability of the model[43]. As detailed in Eq. (36) at the end of this subsection, the † over the softmax operator indicates that the softmax used in Eq. (32) is modified to guarantee smoothness.

We notice that it is fully valid to update the rotationally equivariant representation h_{ij} in a similar way, e.g.,

$$h^{2,l+1}_{ij} = \frac{1}{\sqrt{2}}\left(h^{2,l}_{ij} + \underset{h}{\mathrm{linear}}\left(\sum_{k\in N_{r^1_c}(i)} A^h_{ijk}h^{2,l}_{ik}\right)\right). \tag{33}$$

However, we find such an update would not improve the accuracy and often make the training procedure unstable. Therefore, we choose not to update h_{ij} in the current version of the DPA-2 model.

The smoothness of the softmax operation. The standard softmax is defined by

$$\mathrm{softmax}(x_{ij}) = \frac{e^{x_{ij}}}{\sum_k e^{x_{ik}}}, \tag{34}$$

which introduces a discontinuity in the attention maps in Eqs. (28) and (32). Simply multiplying a switch to the attention maps does not fix the problem. Suppose that one atom comes into the cut-off; the denominator of Eq. (34) changes in a discontinuous way, thus all softmax(x_{ij}) change discontinuously, no matter whether j is the new neighbor or not.

To fix this issue, we define the softmax* by

$$\mathrm{softmax}^*(x_{ij}) = w_{ij}\,\mathrm{softmax}\left(w_{ij}(x_{ij}+s^*) - s^*\right). \tag{35}$$

Similarly, the softmax† is given by

$$\mathrm{softmax}^\dagger(y_{ijk}) = w_{ij}w_{ik}\,\mathrm{softmax}\left(w_{ij}w_{ik}(y_{ijk}+s^\dagger) - s^\dagger\right). \tag{36}$$

It is assumed that the shifting constants s^* and $s^\dagger$ are chosen a magnitude larger than x_{ij} and y_{ijk}, respectively. In practice, the magnitude of both x_{ij} and y_{ijk} in Eqs. (35) and (36) are of order 1, so we set $s^* = s^\dagger = 20$.

Connection to DPA-1 and graph neural network potential models.

The DPA-2 model is closely related to the DPA-1 model[43]. The repinit layer essentially serves as the backbone architecture of the DPA-1. The key difference is that the DPA-1 model applies one or multiple gated attention layers to the pair-atom representation g_{ij}^{rt}, whereas the DPA-2 model stacks multiple repformer layers on top of the repinit layer.

The core innovation of the DPA-2 model lies in its repformer layers, which can be interpreted as an E(3) equivariant graph neural network (GNN). The single-atom representation f_i serves as the node feature, while the pair-atom representations g_{ij} and h_{ij} function as rotationally invariant and equivariant edge features, respectively. Unlike GNNs such as SchNet[53] and NequiP[59], which update node features through convolution and self-interactions, the DPA-2 model employs additional mechanisms for node feature updates, as detailed in Equations (24) and (25). Moreover, DPA-2 enhances edge feature updates within each repformer layer through non-linear self-interaction, a product of node features, and gated self-attention mechanisms, thereby offering greater capacity compared to conventional GNNs.

Single-task training

Suppose that we have a training dataset T of size M, and denote the DFT-labeled energy and force for any configuration $\mathcal{X}_m$, $1 \le m \le M$, by E_m^* and $\{F_{i,m}^*\}$, respectively. The dataset T yields

$$T = \{(\mathcal{X}_1, E_1^*, \{F_{i,1}^*\}), \ldots, (\mathcal{X}_M, E_M^*, \{F_{i,M}^*\})\}. \tag{37}$$

We denote the trainable parameters of the descriptor by θ, and those of the fitting network by ξ. When necessary, the parameters are placed as superscripts of the corresponding notation, i.e., we have $\mathcal{D}_i^\theta$ and $\mathcal{F}^\xi$ for the descriptor and fitting network, respectively. The PES model is thus rewritten as $E = E^{\theta,\xi}(\mathcal{X})$. The loss function at training step t is written as

$$\mathcal{L}(\theta, \xi, B, t) = \frac{1}{|B|}\sum_{m\in B}\left(\frac{p_e(t)}{N}|\Delta E_m^{\theta,\xi}|^2 + \frac{p_f(t)}{3N}\sum_i |\Delta F_{i,m}^{\theta,\xi}|^2\right), \tag{38}$$

$$\Delta E_m^{\theta,\xi} = E^{\theta,\xi}(\mathcal{X}_m) - E_m^*, \tag{39}$$

$$\Delta F_{i,m}^{\theta,\xi} = F_i^{\theta,\xi}(\mathcal{X}_m) - F_{i,m}^*, \tag{40}$$

where B, a randomly sampled subset of $\{1, \ldots, M\}$, represents the minibatch of the training dataset. $p_e(t)$ and $p_f(t)$ are the energy and force prefactors, respectively. If the learning rate at step t is denoted by $\gamma(t)$, then the prefactors are defined by

$$p_\xi(t) = p_\xi^{\text{start}}\frac{\gamma(t)}{\gamma(0)} + p_\xi^{\text{limit}}\left(1 - \frac{\gamma(t)}{\gamma(0)}\right), \quad \xi \in \{e, f\}. \tag{41}$$

At the beginning of the training, the prefactor p_ξ is set to a hyperparameter p_ξ^{start}, and it linearly decays with respect to the learning rate. If the learning rate decays to zero, i.e., $\lim_{t\to\infty}\gamma(t) = 0$, the prefactor converges to the hyperparameter p_ξ^{limit} at the infinite training step. We have adopted the Adam stochastic gradient descent method[92] to minimize the loss function with respect to the model parameters θ and ξ. Virial errors, which are omitted here, can be added to the loss for training if available.

Multi-task training protocol

For various datasets labeled with different DFT calculation parameters, it is infeasible to merge them directly into a single training set for model training. However, these DFT datasets should inherently share a significant amount of commonality, and we expect they can mutually promote each other's training, thus benefiting the overall model capacity.

In this work, to fully utilize various sources of DFT-calculated data, we propose a novel *multi-task* training strategy using a unified model framework for simultaneous training on data calculated with different DFT parameters, as illustrated in Fig. 1a. We first group all the training data into K training datasets, denoted as $\mathcal{T} = \{T_1, \ldots, T_K\}$, where each dataset contains configurations labeled with identical DFT parameters. The configurations and labels in the k-th training dataset are represented by:

$$T_k = \{(\mathcal{X}_{k1}, E_{k1}^*, \{F_{i,k1}^*\}), \ldots, (\mathcal{X}_{kM}, E_{kM}^*, \{F_{i,kM}^*\})\}. \tag{42}$$

We establish a DPA-2 model with the unified descriptor and K fitting networks, and the k-th model is given by:

$$E = E^{\theta,\xi_k}(\mathcal{X}), \tag{43}$$

where ξ_k represents the network parameters of the k-th fitting network. The k-th fitting network is trained by the k-th training dataset, while the unified descriptor (with parameters θ) is *simultaneously* trained by all datasets, and the loss function is given by

$$\mathcal{L}(\theta, \{\xi_k\}, S, \{B\}, t) = \frac{1}{|S|}\sum_{k\in S}\frac{1}{|B_k|}\sum_{m\in B_k}\left(\frac{p_e(t)}{N_m}|\Delta E_{km}^{\theta,\xi_k}|^2 + \frac{p_f(t)}{3N_m}\sum_i |\Delta F_{i,km}^{\theta,\xi_k}|^2\right), \tag{44}$$

$$\Delta E_{km}^{\theta,\xi_k} = E^{\theta,\xi_k}(\mathcal{X}_{km}) - E_{km}^*, \tag{45}$$

$$\Delta F_{i,km}^{\theta,\xi_k} = F_i^{\theta,\xi_k}(\mathcal{X}_{km}) - F_{i,km}^*. \tag{46}$$

At each training step, a subset of the training datasets is sampled $\mathcal{T}$, and the indices of the sampled datasets are denoted by S. B_k represents the minibatch of the training dataset T_k. It should be noted that there is a significant degree of freedom in designing the sampling strategy for S. Sampling can be conducted with a uniform probability or with a bias towards certain systems. Furthermore, sampling may be performed with or without replacement. In our implementation, larger and more complex datasets are assigned a higher probability, and sampling with replacement is employed.

Pre-training and fine-tuning

By utilizing multi-task training on all available training datasets, the configurational and elemental knowledge shared among the datasets is expected to be encoded in the descriptor $\mathcal{D}^{\theta_p}$, with θ_p denoting the converged model parameters. The fitting networks are expected to encode system-specific knowledge. The multi-task training scheme provides the possibility of training with a large number of training datasets (most likely labeled with distinct DFT parameters). Therefore, when trained with a sufficiently large dataset that covers a wide range of configurations and elements for future applications, it is expected that much less training data would be needed to train a new system with the help of the encoded knowledge. The multi-task *pre-trained* model can be used to improve the accuracy and data efficiency in *downstream tasks*. It is worth noting that the downstream task can be either constructing a PES, or a property prediction task, and in this work, we only discuss the PES as a downstream task. The procedure of training a model for downstream tasks from a pre-trained model is called *fine-tuning*.

Given a downstream task training dataset, we may initialize the descriptor of our downstream task model with θ_p to boost the performance compared to a random initialization of the descriptor parameters. Furthermore, if the downstream dataset shares similar configurational and elemental information with any of the fitting networks, then the fitting network of the model could also be initialized with the pre-trained fitting network. The energy bias of the downstream task is determined by the downstream training dataset, rather than by those used in the pre-training stage.

Model distillation

The fine-tuned model possesses a large number of parameters, which might result in low efficiency when directly applied to production scenarios, such as MD simulations. To mitigate this issue, we can distill the model into a more compact version that maintains accuracy on downstream tasks while concurrently achieving speed enhancements and enabling large-scale simulations. The distillation process, illustrated in Fig. 1c, consists of an iterative concurrent learning loop. The model prior to distillation, denoted as the teacher model, is used for data labeling, whereas a student model featuring a simpler model structure (e.g., DPA-1 without any attention layer, which can be further compressed[78] to significantly enhance performance) is trained on the labeled data. Subsequently, the teacher model is utilized for MD exploration, adopting simulation settings similar to those of downstream tasks, ensuring that the elemental and configurational spaces explored during distillation and downstream tasks exhibit overlap. Configurations are sampled from the simulated MD trajectories, and the inference deviations between the teacher and student models on those samples are assessed. Samples with model deviation exceeding a predetermined threshold are added to the training dataset for the next iteration. This procedure is repeated until the student model's accuracy satisfies our criteria or no longer changes.

Data availability

The datasets and models used in this study, as detailed in the section S1 of the Supplementary Materials, are all available on AIS Square (https://www.aissquare.com). The codes, datasets, and input scripts are all available on zenodo (https://doi.org/10.5281/zenodo.13342300). Finally, to test the models, users are welcome to consider going through this Bohrium Notebook (https://nb.bohrium.dp.tech/detail/18475433825), and explore the DP Combo web server (https://app.bohrium.dp.tech/dp-combo).

Received: 19 August 2024; Accepted: 27 November 2024;
Published online: 19 December 2024

References

1. Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Phys. Rev.* **136**, B864 (1964).
2. Kohn, W. & Sham, Lu. Jeu Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **140**, A1133 (1965).
3. Badaoui, M. et al. Combined free-energy calculation and machine learning methods for understanding ligand unbinding kinetics. *J. Chem. Theory Comput.* **18**, 2543–2555 (2022).
4. Zeng, J., Tao, Y., Giese, T. J. & York, D. M. Qdπ: a quantum deep potential interaction model for drug discovery. *J. Chem. Theory Comput.* **19**, 1261–1275 (2023).
5. Bartók, A. P., Kermode, J., Bernstein, N. & Csányi, G. ábor Machine learning a general-purpose interatomic potential for silicon. *Phys. Rev. X* **8**, 041048 (2018).
6. Deringer, V. L., Caro, M. A. & Csányi, G. ábor A general-purpose machine-learning force field for bulk and nanostructured phosphorus. *Nat. Commun.* **11**, 1–11 (2020).
7. Wen, T., Zhang, L., Wang, H., Weinan, E. & Srolovitz, D. J. Deep potentials for materials science. *Mater. Futures* **1**, 022601 (2022).
8. Ma, S. & Liu, Zhi-Pan Machine learning for atomic simulation and activity prediction in heterogeneous catalysis: current status and future. *ACS Catal.* **10**, 13213–13226 (2020).
9. Yang, M., Raucci, U., & Parrinello, M. Ammonia decomposition on lithium imide surfaces: a new paradigm in heterogeneous catalysis. *ChemRxiv* https://doi.org/10.26434/chemrxiv-2022-qr7wt (2022).
10. Car, R. & Parrinello, M. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.* **55**, 2471 (1985).
11. Smith, J. S., Nebgen, B., Lubbers, N., Isayev, O. & Roitberg, A. E. Less is more: sampling chemical space with active learning. *J. Chem. Phys.* **148**, 241733 (2018).
12. Uteva, E., Graham, R. S., Wilkinson, R. D. & Wheatley, R. J. Active learning in gaussian process interpolation of potential energy surfaces. *J. Chem. Phys.* **149**, 174114 (2018).
13. Zhang, L., Lin, De-Ye, Wang, H., Car, R. & Weinan, E. Active learning of uniformly accurate interatomic potentials for materials simulation. *Phys. Rev. Mater.* **3**, 023804 (2019).
14. Zhang, Y. et al. Dp-gen: a concurrent learning platform for the generation of reliable deep learning based potential energy models. *Comput. Phys. Commun.* **253**, 107206 (2020).
15. Jiang, W., Zhang, Y., Zhang, L. & Wang, H. Accurate deep potential model for the Al–Cu–Mg alloy in the full concentration space. *Chin. Phys. B* **30**, 050706 (2021).
16. Takamoto, S. et al. Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements. *Nat. Commun.* **13**, 2991 (2022).
17. Chen, C. & Ong, Shyue Ping. A universal graph deep learning interatomic potential for the periodic table. *Nat. Comput. Sci.* **2**, 718–728 (2022).
18. Choudhary, K. et al. Unified graph neural network force-field for the periodic table: solid state applications. *Digit. Discov.* **2**, 346–355 (2023).
19. Deng, B. et al. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nat. Mach. Intell.* **5**, 1031–1041 (2023).
20. Batatia, I. et al. A foundation model for atomistic materials chemistry. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2401.00096 (2023).
21. Merchant, A. et al. Scaling deep learning for materials discovery. *Nature* **624**, 1–6 (2023).
22. Jain, A. et al. The materials project: a materials genome approach to accelerating materials innovation. APL Mater **1**, 011002 (2013).
23. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865 (1996).
24. Chai, J.-D. & Head-Gordon, M. Systematic optimization of long-range corrected hybrid density functionals. *J. Chem. Phys*. **128**, 084106 (2008).
25. Cui, T. et al. Gpip: geometry-enhanced pre-training on interatomic potentials. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2309.15718 (2023).
26. Feng, R. et al. May the force be with you: unified force-centric pre-training for 3d molecular conformations. *Adv. Neural Inf. Process. Syst.* **36**, (2024).
27. Jacobson, L., Stevenson, J., Ramezanghorbani, F., Dajnowicz, S., & Leswing, K. Leveraging multitask learning to improve the transferability of machine learned force fields. *ChemRxiv*, (2023).
28. Wang, Y., Xu, C., Li, Z. & Farimani, A. B. Denoise pre-training on non-equilibriummolecules for accurate and transferable neuralpotentials. *J. Chem. Theory Comput.* **19**, 5077–5087 (2023).
29. Kokkinos, I. Ubernet: training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of Proceedings of the IEEE conference on computer vision and pattern recognition* 6129–6138 (CVPR, 2017).
30. Devlin, J., Chang, Ming-Wei, Lee, K., & Toutanova, K. Bert: pre-training of deep bidirectional transformers for language understanding. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1810.04805 (2018).
31. Shoghi, N. et al. From molecules to materials: pre-training large generalizable models for atomic property prediction. In *Proceedings of The Twelfth International Conference on Learning Representations*, (2024).
32. Smith, J. S. et al. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nat. Commun.* **10**, 1–8 (2019).
33. Kolluru, A. et al. Transfer learning using attentions across atomic systems with graph neural networks (taag). *J. Chem. Phys.* **156**, 184702 (2022).

34. Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning* (MIT Press, 2016).
35. Tuckerman, M. *Statistical Mechanics: Theory and Molecular Simulation* (OUP Oxford, 2010).
36. Behler, J. örg & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
37. Smith, J. S., Isayev, O. & Roitberg, A. E. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).
38. Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 1–8 (2017).
39. Gastegger, M., Schwiedrzik, L., Bittermann, M., Berzsenyi, F., & Marquetand, P. wACSF–weighted atom-centered symmetry functions as descriptors in machine learning potentials. *J. Chem. Phys*. **148**, 241709 (2018).
40. Zhang, L., Han, J., Wang, H., Car, R. & Weinan, E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **120**, 143001 (2018).
41. Zhang, L. et al. End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems. *Adv. Neural Inf. Process. Syst*. **31**, 4441–4451 (2018).
42. Zeng, J. et al. Deepmd-kit v2: a software package for deep potential models. *J. Chem. Phys.* **159**, 054801 (2023).
43. Zhang, D. et al. Pretraining of attention-based deep learning potential model for molecular simulation. *npj Comput. Mater.* **10**, 94 (2024).
44. Zhang, Y., Hu, C. & Jiang, B. Embedded atom neural network potentials: Efficient and accurate machine learning with a physically inspired representation. *J. Phys. Chem. Lett.* **10**, 4962–4967 (2019).
45. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. ábor Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **104**, 136403 (2010).
46. Bartók, A. P., Kondor, R. & Csányi, G. ábor On representing chemical environments. *Phys. Rev. B* **87**, 184115 (2013).
47. Rupp, M., Tkatchenko, A., Müller, Klaus-Robert & Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301 (2012).
48. Chmiela, S. et al. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **3**, e1603015 (2017).
49. Thompson, A. P., Swiler, L. P., Trott, C. R., Foiles, S. M. & Tucker, G. J. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *J. Comput. Phys.* **285**, 316–330 (2015).
50. Shapeev, A. V. Moment tensor potentials: a class of systematically improvable interatomic potentials. *Multiscale Model. Simul.* **14**, 1153–1173 (2016).
51. Thomas, N. et al. Tensor field networks: rotation-and translation-equivariant neural networks for 3d point clouds. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1802.08219 (2018).
52. Batzner, S. et al. Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 1–11 (2022).
53. Schütt, K. et al. Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. *Adv. Neural Inf. Process. Syst*. **30**, 992–1002 (2017).
54. Gasteiger, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2011.14115 (2020).
55. Schütt, K. T., Unke, O. T. & Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *Proceedings of International Conference on Machine Learning,* 9377–9388 (PMLR, 2021).
56. Gasteiger, J., Becker, F. & Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Adv. Neural Inf. Process. Syst.* **34**, 6790–6802 (2021).
57. Shuaibi, M. et al. Rotation invariant graph neural networks using spin convolutions. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2106.09575 (2021).
58. Zitnick, L. et al. Spherical channels for modeling atomic interactions. *Adv. Neural Inf. Process. Syst.* **35**, 8054–8067 (2022).
59. Batzner, S. et al. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 1–11 (2022).
60. Batatia, I., Kovacs, D. P., Simm, G., Ortner, C. & Csányi, G. ábor Mace: higher order equivariant message passing neural networks for fast and accurate force fields. *Adv. Neural Inf. Process. Syst.* **35**, 11423–11436 (2022).
61. Liao, Y.-L. & Smidt, T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2206.11990 (2022).
62. Liao, Y.-L., Wood, B., Das, A., & Smidt, T. Equiformerv2: improved equivariant transformer for scaling to higher-degree representations. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2306.12059 (2023).
63. Musaelian, A. et al. Learning local equivariant representations for large-scale atomistic dynamics. *Nat. Commun.* **14**, 579 (2023).
64. Bengio, Y., Courville, A. & Vincent, P. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798–1828 (2013).
65. Zhang, D., Yin, J., Zhu, X. & Zhang, C. Network representation learning: a survey. *IEEE Trans. Big Data* **6**, 3–28 (2018).
66. Radford, A. et al. *Improving Language Understanding by Generative Pre-training* (*OpenAI Blog*, 2018).
67. Dosovitskiy, A. et al. An image is worth 16 × 16 words: transformers for image recognition at scale. In *Proceedings of International Conference on Learning Representations*, (2021).
68. Zhou, G. et al. Uni-mol: a universal 3D molecular representation learning framework. *ChemRxiv* https://doi.org/10.26434/chemrxiv-2022-jjm0j-v4 (2022).
69. Zhu, J. et al. Unified 2D and 3D pre-training of molecular representations. In *Proceedings of Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2626–2636 (ACM, 2022).
70. Zaidi, S. et al. Pre-training via denoisingfor molecular property prediction. In *Proceedings of NeurIPS 2022 AI for Science: Progress and Promises*, (2022).
71. Lu, S., Gao, Z., He, D., Zhang, L., & Ke, G. Highly accurate quantum chemical propertyprediction with uni-mol+. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2303.16982 (2023).
72. Feng, S., Ni, Y., Lan, Y., Ma, Z. & Ma, W.-Y. Fractional denoising for 3D molecular pre-training. In *Proceedings of International Conference on Machine Learning*, 9938–9961 (PMLR, 2023).
73. Jiao, R., Han, J., Huang, W., Rong, Y. & Liu, Y. Energy-motivated equivariant pretraining for 3d molecular graphs. In *Proceedings of Proceedings of the AAAI Conference on Artificial Intelligence*, 378096–8104 (2023).
74. Beaini, D. et.al. Towards foundational models for molecular learning on large-scale multi-task datasets. In *Proceedings of The Twelfth International Conference on Learning Representations*, (2024).
75. Lee, K. L. K. et al. Towards foundation models for materials science: The open matsci ml toolkit. In *Proceedings of Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, 51–59 (2023).
76. Gardner, John LA, Baker, K. T., & Deringer, V. L. Synthetic pre-training for neural-network interatomic potentials. *Mach. Learn. Sci. Technol.* https://doi.org/10.1088/2632-2153/ad1626 (2023).

77. Chanussot, L. et al. Open catalyst 2020 (OC20) dataset and community challenges. *ACS Catal.* **11**, 6059–6072 (2021).
78. Lu, D. et al. Dp compress: a model compression scheme for generating efficient deep potential models. *J. Chem. Theory Comput.* **18**, 5559–5567 (2022).
79. Kresse, G. & Furthmüller, J. ürgen Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **6**, 15–50 (1996).
80. Kresse, G. & Furthmüller, J. ürgen Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54**, 11169 (1996).
81. Frisch, M.J. et al. *Gaussian 16, Revision B.01* (Gaussian, Inc., 2016).
82. Chen, M., Guo, G. C. & He, L. Systematically improvable optimized atomic basis sets for ab initio calculations. *J. Phys. Condens. Matter* **22**, 445501 (2010).
83. Li, P. et al. Large-scale ab initio simulations based on systematically improvable atomic basis. *Comput. Mater. Sci.* **112**, 503–517 (2016).
84. Gasteiger, J. et al. Gemnet-OC: Developing graph neural networks for large and diverse molecular simulation datasets. *Transactions on Machine Learning Research*, (2022).
85. Wu, J. et al. Universal interatomic potential for perovskite oxides. *Phys. Rev. B* **108**, L180104 (2023).
86. Mo, Y., Ong, ShyuePing & Ceder, G. First principles study of the li10gep2s12 lithium super ionic conductor material. *Chem. Mater.* **24**, 15–17 (2012).
87. Marcolongo, A. & Marzari, N. Ionic correlations and failure of nernst-einstein relation in solid-state electrolytes. *Phys. Rev. Mater.* **1**, 025402 (2017).
88. Kuhn, A., Köhler, J. ürgen & Lotsch, B. V. Single-crystal X-ray structure analysis of the superionic conductor li 10 gep 2 s 12. *Phys. Chem. Chem. Phys.* **15**, 11620–11622 (2013).
89. Huang, J. et al. Deep potential generation scheme and simulation protocol for the li10gep2s12-type superionic conductors. *J. Chem. Phys.* **154**, 094703 (2021).
90. Wang, D., Cao, M. & Zhang, S. Phase diagram and properties of Pb (In1/2Nb1/2) O3–Pb (Mg1/3Nb2/3) O3–PbTiO3 polycrystalline ceramics. *J. Eur. Ceram. Soc.* **32**, 433–439 (2012).
91. Li, Q. et al. Soft phonon modes and diffuse scattering in Pb (In1/2Nb1/2) O3–Pb (Mg1/3Nb2/3) O3–PbTiO3 relaxor ferroelectrics. *J. Mater.* **4**, 345–352 (2018).
92. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1412.6980 (2014).
93. DiStasio, R. A., Santra, B., Li, Z., Wu, X., & Car, R. The individual and collective effects of exact exchange and dispersion interactions on the ab initio structure of liquid water. *J. Chem. Phys*. **141**, 084502 (2014).

Acknowledgements

We gratefully acknowledge the support received for this work. The work of Han Wang is supported by the National Key R&D Program of China (grant no. 2022YFA1004300) and the National Natural Science Foundation of China (grant no. 12122103). The work of Weinan E is supported by the National Key Research and Development Project of China (grant no. 2022YFA1004302) and the National Natural Science Foundation of China (grants nos. 92270001 and 12288101). The work of Jinzhe Zeng and Darrin M. York is supported by the National Institutes of Health (grant no. GM107485 to D.M.Y.) and the National Science Foundation (grant no. 2209718 to D.M.Y.). The work of Shi Liu is supported by the Natural Science Foundation of Zhejiang Province (grant no. 2022XHSJJ006). The work of Tong Zhu is supported by the National Natural Science Foundation of China (grants nos. 22222303 and 22173032). The work of Zhicheng Zhong is supported by the National Key R&D Program of China (grants nos. 2021YFA0718900 and 2022YFA1403000). The work of Jian Lv is supported by the National Natural Science Foundation of China (grants nos. 12034009 and 91961204). The work of Jun Cheng is supported by the National Science Fund for Distinguished Young Scholars (grant no. 22225302), Laboratory of AI for Electrochemistry (AI4EC), and IKKEM (grants nos. RD2023100101 and RD2022070501). The work of Mohan Chen is supported by the National Natural Science Foundation of China (grants nos. 12122401, 12074007, and 12135002). Lastly, the computational resource was supported by the Bohrium Cloud Platform at DP Technology and Tan Kah Kee Supercomputing Center (IKKEM).

Author contributions

D.Z., X.L., H.W., and L.Z. conceived the idea of this work, designed the model structure, and implemented the model. The experiments were mainly designed and performed by D.Z., X.L., X.Z., C.Z., C.C., H.B., Y.D., X.Q., and A.P. Other authors performed data collection and model tests on different systems. All authors contributed to the discussions and edited the manuscript.

Competing interests

The authors declare no competing financial or non-financial interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-024-01493-2.

Correspondence and requests for materials should be addressed to Linfeng Zhang or Han Wang.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

[1]AI for Science Institute, Beijing, P. R. China. [2]DP Technology, Beijing, P. R. China. [3]Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, P. R. China. [4]State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P.R. China. [5]University of Chinese Academy of Sciences, Beijing, P.R. China. [6]HEDPS, CAPT, College of Engineering, Peking University, Beijing, P.R. China. [7]Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo, P.R. China. [8]CAS Key Laboratory of Magnetic Materials and Devices and Zhejiang Province Key Laboratory of Magnetic Materials and Application Technology, Chinese Academy of Sciences, Ningbo, P.R. China. [9]School of Electronics Engineering and Computer Science, Peking University, Beijing, P.R. China. [10]Shanghai Engineering Research Center of Molecular Therapeutics & New Drug Development, School of Chemistry and Molecular Engineering, East China Normal University, Shanghai, P.R. China. [11]Laboratory for Biomolecular Simulation Research, Institute for Quantitative Biomedicine and Department of Chemistry and Chemical Biology, Rutgers University, Piscataway, NJ, USA. [12]Department of Chemistry, Princeton University, Princeton, NJ, USA. [13]College of Chemistry and Molecular Engineering, Peking University, Beijing, P.R. China. [14]Yuanpei College, Peking University, Beijing, P.R. China. [15]School of Electrical Engineering and Electronic Information, Xihua University, Chengdu, P.R. China. [16]State Key Laboratory of Superhard Materials, College of Physics, Jilin University, Changchun, P.R. China. [17]Key Laboratory of Material Simulation Methods & Software of Ministry of Education, College of Physics, Jilin University, Changchun, P.R. China. [18]International Center of Future Science, Jilin University, Changchun, P.R. China. [19]Key Laboratory for Quantum Materials of Zhejiang Province, Department of Physics, School of Science, Westlake University, Hangzhou, P.R. China. [20]Atomistic Simulations, Italian Institute of Technology, Genova, Italy. [21]State Key Laboratory of Physical Chemistry of Solid Surface, iChEM, College of Chemistry and Chemical Engineering, Xiamen University, Xiamen, P.R. China. [22]Institute of Natural Sciences, Westlake Institute for Advanced Study, Hangzhou, P.R. China. [23]NYU-ECNU Center for Computational Chemistry at NYU Shanghai, Shanghai, P.R. China. [24]Institute for Advanced Algorithms Research, Shanghai, P.R. China. [25]Laboratory of AI for Electrochemistry (AI4EC), IKKEM, Xiamen, P.R. China. [26]Institute of Artificial Intelligence, Xiamen University, Xiamen, P.R. China. [27]Center for Machine Learning Research, Peking University, Beijing, P.R. China. [28]School of Mathematical Sciences, Peking University, Beijing,, P.R. China. [29]Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing, P.R. China. [30]These authors contributed equally: Duo Zhang, Xinzijian Liu. ✉e-mail: linfeng.zhang.zlf@gmail.com; wang_han@iapcm.ac.cn