Graph neural network model for the era of large atomistic models

Duo Zhang[1,2,3,*], Anyang Peng[1], Chun Cai[1,2], Wentao Li[4], Yuanchang Zhou[5,6], Jinzhe Zeng[7,8,9], Mingyu Guo[1,2,10], Chengqian Zhang[1,2,3], Bowen Li[11], Hong Jiang[12], Tong Zhu[11,13,14], Weile Jia[5,6], Linfeng Zhang[1,2,†], and Han Wang[15,16,‡]

[1]AI for Science Institute, Beijing 100080, P. R. China
[2]DP Technology, Beijing 100080, P. R. China
[3]Academy for Advanced Interdisciplinary Studies, Peking University, Beijing 100871, P. R. China
[4]Department of Chemical Engineering, Tsinghua University, Beijing 100084, P. R. China.
[5]State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100871, P.R. China
[6]University of Chinese Academy of Sciences, Beijing 100871, P.R. China
[7]School of Artificial Intelligence and Data Science, University of Science and Technology of China, Hefei 230026, P. R. China
[8]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou 215123, P. R. China
[9]Suzhou Big Data & AI Research and Engineering Center, Suzhou 215123, P. R. China
[10]School of Chemistry, Sun Yat-sen University, Guangzhou, P. R. China
[11]Shanghai Engineering Research Center of Molecular Therapeutics & New Drug Development, School of Chemistry and Molecular Engineering, East China Normal University, Shanghai 200062, P.R. China
[12]College of Chemistry and Molecular Engineering, Peking University Beijing 100871, P. R. China
[13]NYU-ECNU Center for Computational Chemistry at NYU Shanghai, Shanghai 200062, P.R. China
[14]Institute for Advanced algorithms research, Shanghai, 201306, P.R. China
[15]National Key Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Fenghao East Road 2, Beijing 100094, P.R. China
[16]HEDPS, CAPT, College of Engineering, Peking University, Beijing 100871, P.R. China

ABSTRACT

Foundation models, or large atomistic models (LAMs), aim to universally represent the ground-state potential energy surface (PES) of atomistic systems as defined by density functional theory (DFT). The scaling law is pivotal in the development of large models, suggesting that their generalizability in downstream tasks consistently improves with increased model size, expanded training datasets, and larger computational budgets. In this study, we present DPA3, a multi-layer graph neural network founded on line graph series (LiGS), designed explicitly for the era of LAMs. We demonstrate that the generalization error of the DPA3 model adheres to the scaling law. The scalability in the number of model parameters is attained by stacking additional layers within DPA3. Additionally, the model employs a dataset encoding mechanism that decouples the scaling of training data size from the model size within its multi-task training framework. When trained as problem-oriented potential energy models, the DPA3 model exhibits superior accuracy in the majority of benchmark cases, encompassing systems with diverse features, including molecules, bulk materials, surface and cluster catalysis, two-dimensional materials, and battery materials. When trained as a LAM on the OpenLAM-v1 dataset, the DPA-3.1-3M model exhibits state-of-the-art performance in the LAMBench benchmark suit for LAMs, demonstrating lowest overall zero-shot generalization error across 17 downstream tasks from a broad spectrum of research domains. This performance suggests superior accuracy as an out-of-the-box potential model, requiring minimal fine-tuning data for downstream scientific applications.

1 Introduction

The foundational model for atomistic systems is predicated on the Schrödinger equation[1], with the assumption that relativistic effects are negligible. The ground state solution to the Schrödinger equation, applying the Born-Oppenheimer approximation[2], defines a universal potential energy surface (PES), which plays a central role in the computational simulation of systems at the atomistic scale[3]. In practice, Kohn-Sham density functional theory (DFT)[4,5] is frequently employed as a computationally feasible approximation to the ground state Schrödinger equation. While DFT offers an attractive balance between accuracy and efficiency for most applications, the computational demand remains significant, as the complexity scales cubically with the electronic degrees of freedom.

Over the past decade, machine learning interatomic potentials (MLIPs)[6–12] have emerged as an efficient surrogate for DFT calculations, significantly reducing computational costs to linear scaling while maintaining comparable accuracy[13]. MLIPs are typically trained to address specific scientific challenges. However, when investigating a new system, the model must be re-parameterized, necessitating a substantial amount of DFT calculations to label the training data. This requirement has stimulated the development of universal models or large atomistic models (LAMs), which aim to universally represent the ground-state potential energy surface of the atomistic systems[14–22]. The feasibility of LAMs is grounded in the universality of the DFT solution. These models are expected to be used as out-of-the-box potential energy surfaces[19] or fine-tuned with substantially less training data[23–25] for downstream tasks.

While some LAMs have been successfully applied, yielding substantial advances in research—such as the GNoME model, which discovered 381,000 new stable structures—a considerable disparity remains in the generalizability of state-of-the-art LAMs compared to MLIPs specifically trained for particular problems[26]. To bridge this gap, enhancements in LAM architectures are expected to focus on improving generalizability across diverse research domains. Additionally, the development of LAMs is anticipated to adhere to scaling laws[27–30], which propose that the generalizability of these models can be improved by systematically expanding the dataset size, model parameters, and computational budgets. Despite the pivotal role of scaling laws in model development, their exploration within LAMs remains limited. Notably, GNoME demonstrated a scaling law with respect to the number of training data, while Uni-Mol2[31], a pretrained model for molecular sciences, illustrated scaling laws concerning data, parameters, and computational budgets. However, a recent study also reveals the non-trivial nature of scaling law, particularly for GNN-based architectures where oversmoothing effects pose significant challenges[32]. The specific model architecture through which a LAM may exhibit these scaling laws remains an open question.

According to the scaling laws, a key strategy for enhancing the generalizability of LAMs is to incorporate extensive training datasets that span a broad spectrum of research domains. However, inconsistencies arise due to variations in exchange-correlation (XC) functionals, discretization basis sets, and software implementations in DFT calculations, rendering available training data incompatible for merging to train LAMs. A potential solution is to expand existing datasets with compatible DFT calculation settings. For example, the Open Materials 2024 dataset[33] maintains consistent settings, specifically the PBE/PBE+U[34] functional and planewave basis, with the Materials Project[35], while significantly increasing data volume. However, this extension may not be able to satisfy the requirements for XC functionals and basis sets in other domains, such as small molecules, where hybrid functionals and atomic basis sets are commonly used[36–38]. Alternatively, multi-task training offers a method to learn common knowledge across a wide ar-

*zhduodyx@pku.edu.cn
†linfeng.zhang.zlf@gmail.com
‡wang_han@iapcm.ac.cn

ray of datasets, irrespective of their DFT settings[24]. Nonetheless, this approach faces scalability challenges, as the number of fitting heads needed to learn dataset-specific knowledge increases with the number of datasets. Furthermore, the inconsistent XC preferences across different domains drive the development of multi-fidelity models[39].

The LAMs are expected to produce physically meaningful results in molecular simulations, necessitating adherence to all physical laws inherent to the universal PES. Specifically, LAMs should be smooth and conservative, ensuring energy conservation in microcanonical ensemble molecular dynamics (MD) simulations and maintaining Boltzmann distributions in canonical and isothermal-isobaric ensemble MD simulations[40]. LAMs must also be invariant under translational and rotational transformations, thereby conserving linear and angular momentum, respectively, in accordance with Noether's theorem. Additionally, LAMs should respect the indistinguishability of atoms of the same chemical species, maintaining invariance under their permutations, which is fundamental to quantum statistics. Recent studies indicate that smooth, conservative models demonstrate a higher correlation between force field accuracy and property predictions, a critical factor in downstream applications[41].

In this study, we present the DPA3 model architecture, a message-passing neural network built upon line graph series (LiGS) specifically designed for the forthcoming era of LAMs. The model is designed to exhibits scaling laws, whereby its generalizability improves consistently with increases in model size, the volume of training data, and computational budgets. Furthermore, the DPA3 model incorporates dataset encoding to differentiate between training datasets, enabling multi-task training across diverse datasets irrespective of their DFT settings, with the advantage that the overhead does not scale with the number of datasets. Last but not least, the DPA3 model is rigorously aligned with all physical laws associated with the universal PES.

The advantages of the DPA3 model architecture are firstly demonstrated through MLIP tasks that are well-established in the literature, where DPA3 consistently outperforms state-of-the-art GNN models in most instances. Subsequently, the scaling law of the DPA3 in LAM tasks is validated through training on the MPtrj dataset, which has been utilized to train LAMs such as CHGNet[17] and MACE-MP-0[19]. Furthermore, the DPA3 model is trained on the OpenLAM-v1 dataset[26,42], where the resultant DPA-3.1-3M model exhibits superior performance compared to state-of-the-art LAMs in the force field tasks of the LAMBench benchmark suite, which is intended to demonstrate the generalizability of LAMs in addressing real-world scientific challenges. All of the features position DPA3 as an exceptionally suitable candidate for the era of LAMs.

2 Results

2.1 DPA3: a graph neural network on line graph series

The DPA3 model is a graph neural network that operates on a series of graphs generated through the line graph transformation[43–45]. Given a graph formed by vertices and edges, the line graph transform $\mathscr{L}$ constructs a new graph, denoted as $\mathscr{L}(G)$. This process involves two steps, as illustrated in Fig. 1(a): first, each edge in G becomes a vertex in $\mathscr{L}(G)$; second, an edge is formed between two vertices in $\mathscr{L}(G)$ if their corresponding edges in G share a common vertex. It is important to note that applying the line graph transform to a graph results in another graph. Therefore, starting with an initial graph $G^{(1)}$, one can recursively generate a series of graphs $\{G^{(1)}, G^{(2)}, \ldots, G^{(K)}\}$ using the line graph transform $G^{(k)} = \mathscr{L}(G^{(k-1)})$ for any $1 < k \leq K$. This sequence is referred to as the Line Graph Series (LiGS) generated from the graph $G^{(1)}$, as illustrated by Fig. 1(b). In this series, we call graph $G^{(k)}$ has an order of k, while the maximal order K also defines the order of the LiGS.

In an atomistic system, a graph $G^{(1)}$ can be defined by representing atoms as vertices and pairs of

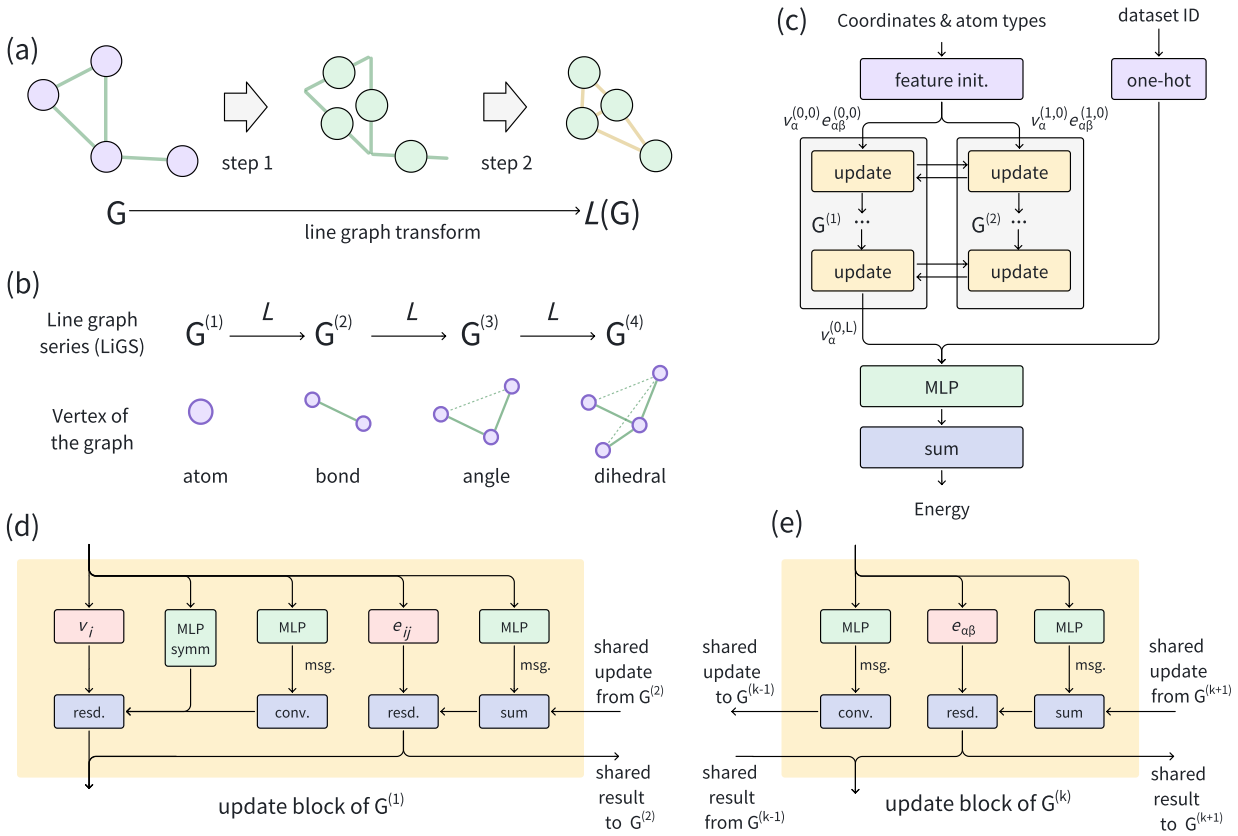

Figure 1. Schematic plot of the DPA3 model architecture. (a) The line graph transform. (b) The line graph series (LiGS). (c) The model architecture of DPA3, a graph neural network on LiGS. (d) The update block of graph $G^{(1)}$. (d) The update block of graph $G^{(k)}$, $k > 1$.

neighboring atoms as edges. The neighborhood of a given atom i consists of all other atoms within a user-defined cutoff radius, r_c^1, from atom i. A correspondence can be established between the vertices and edges in the LiGS and geometric entities within an atomistic system. Specifically, vertices in $G^{(2)}$, $G^{(3)}$, and $G^{(4)}$ correspond to a bond defined by two neighboring atoms, an angle formed by three atoms with two of the bonds sharing a common atom, and a dihedral angle defined by four atoms with two of the angles sharing a common bond, respectively, as illustrated in Fig. 1(b).

The DPA3 model is a multi-layer message-passing neural network defined on the LiGS, as depicted in Fig. 1 (b). In this model, the layer l vertex and edge features on the LiGS graph $G^{(k)}$ are represented by $v_\alpha^{(k,l)} \in \mathbb{R}^{d_v}$ and $e_{\alpha\beta}^{(k,l)} \in \mathbb{R}^{d_e}$, respectively, where α and $\alpha\beta$ denote vertex and edge indices, d_v and d_e denote the dimensionality of the vertex and edge features, respectively. The feature updating process within the LiGS is conducted iteratively across all graphs using a recursive formula. At each layer, vertex features are refined through the convolution of messages transmitted via all edges connecting to the vertex, while edge features are updated based on a message formed from the edge feature itself combined with the features of the two terminal vertices. The updating mechanism employs a residual formulation to ensure model stability, even with the integration of multiple update layers. Importantly, the vertex feature of graph $G^{(k)}$ is identical to the edge feature of the preceding graph $G^{(k-1)}$ within the LiGS framework. This identity eliminates the necessity of preserving redundant vertex feature data for any graph $G^{(k)}$ where $k > 1$. Instead, vertex feature updates are transferred to adjacent graphs to revise the edge features of $G^{(k-1)}$. Subsequently, these updated edge features in graph $G^{(k-1)}$ dictate the vertex features of graph $G^{(k)}$, as illustrated in Fig. 1(e). For the initial graph $G^{(1)}$, vertex features are iteratively refined by incorporating additional self-message and symmetrization transformation, see Fig. 1 (d). The detailed information on the feature udpate scheme is provided in Sec. 4.2.

The final vertex feature of the initial graph $G^{(1)}$ is employed as a descriptor to represent the local environment of any atom within the system. Additionally, it is possible to aggregate vertex features from the graphs in the LiGS using suitable pooling methods. This approach enhances model performance in shallow architectures, whereas a decline in performance is observed with deeper models. Given that model capacity is primarily augmented through deeper networks, we exclusively utilize the vertex feature from the first graph to construct the descriptor.

In the case of multi-task training, the descriptor is further augmented with a dataset encoding, typically represented as a one-hot vector corresponding to the training datasets. This enhanced feature is then input to a fitting MLP to predict the atomic contributions to the energy, which are subsequently combined to determine the total system energy. Forces and virials are derived by back-propagating the DPA3 predicted system energy with respect to atomic coordinates and cell tensors, respectively. Consequently, the DPA3 model is inherently conservative. The proof demonstrating that the DPA3 model is smooth and invariant to translational, rotational, and permutation symmetry operations is provided in Sec. 4.2.

2.2 Benchmarking

The DPA3 model is evaluated using five test cases established in the literature, as depicted in Fig. 2. In each case, the DPA3 models are trained as MLIPs, with a focus on achieving high accuracy in predicting energies, forces, and stress tensors. The results are then compared with those from state-of-the-art MLIP architectures. In these benchmarks, we did not prioritize the accuracy of property calculations, primarily because the DPA3 model is designed to be smooth and conservative, thus convergence in force field error will inherently lead to convergence in property calculation error[41]. In all instances, the maximum model size of the DPA3 is limited by the memory capacity of the GPU hardware accessible to the authors during the training period, specifically the Nvidia A800 GPU, which features 80GB of memory.

We first benchmark DPA3 on the SPICE-MACE-OFF[36], which was developed by Kovács et.al. for

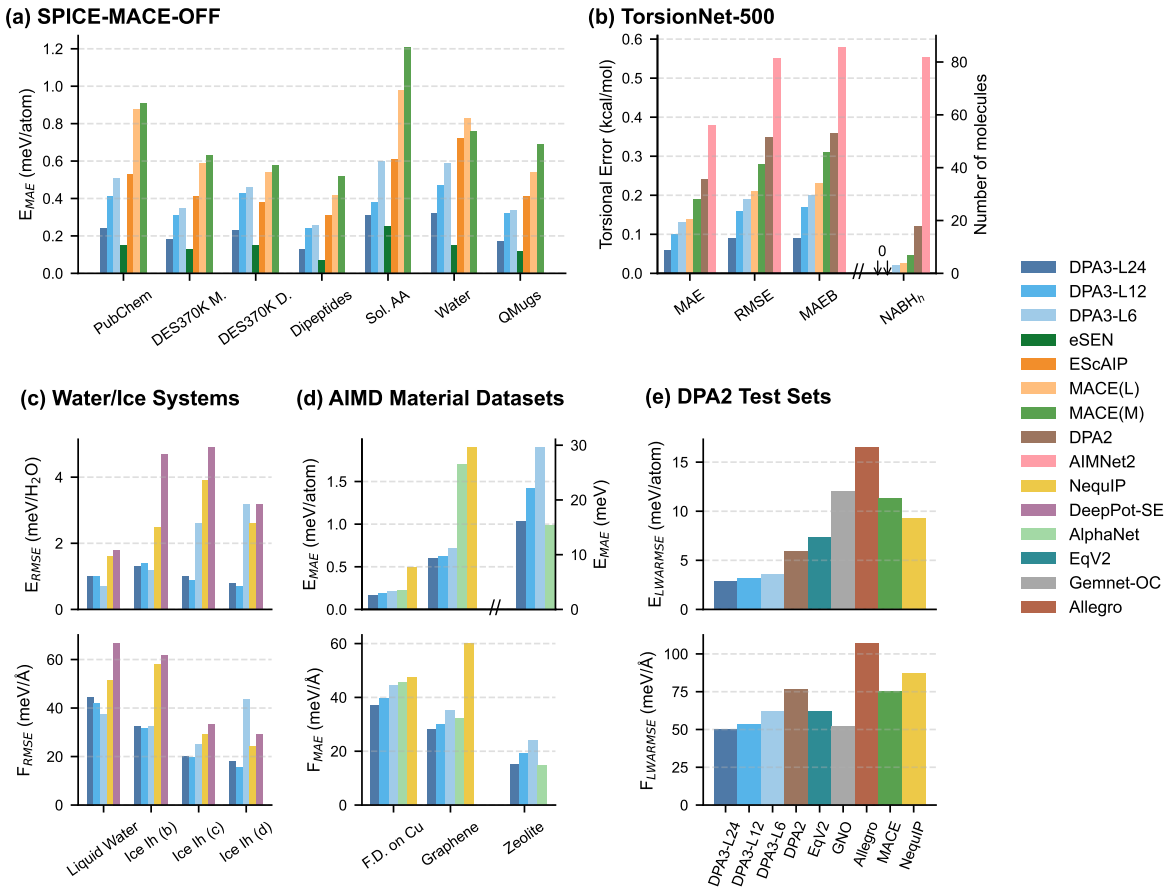

Figure 2. Comparative performance of DPA3 with other MLIPs across different benchmarks. (a) Test energy MAE (E_{MAE}, meV/atom) evaluated on the SPICE-MACE-OFF dataset[36]. Abbreviations: DES370k M. (DES370k Monomers), DES370k D. (DES370k Dimers) and Sol. AA (Solvated Amino Acids). (b) Torsional energy prediction errors quantified by MAE, RMSE, barrier height MAE (MAEB), and the number of accurately predicted barrier heights within 1 kcal/mol ($NABH_h$) on the TorsionNet-500 dataset[46], following evaluation methods described in Ref.[47]. (c) Test energy RMSE (E_{RMSE}, meV/H_2O) and force RMSE (F_{RMSE}, meV/Å) for liquid water and three ice configurations (Ih (b), Ih (c), and Ih (d), sampled at different thermodynamic states)[8]. (d) Test energy MAE (E_{MAE}, meV/atom for the first two datasets and meV for the third) and force MAE (F_{MAE}, meV/Å) evaluated on three distinct material datasets from Ref.[48], namely, the Formate Decomposition on Cu (F.D. on Cu), the Defected Bilayer Graphene and the Zeolite dataset. (e) Logarithmic Weighted Average RMSE of energy ($E_{LWARMSE}$, meV/atom) and force ($F_{LWARMSE}$, meV/Å), as defined in Eq. (1), evaluated on the DPA2 test sets comprising 18 diverse cases detailed in Ref.[24].

training a MLIP for organic small molecule tasks. The dataset is composed by 1M configuraitons (95% for training/validation and 5% for test) with each label with the ωB97M-D3(BJ)/def2-TZVPPD[49–53] XC functional and basis set. We have kept the division of the training, validation and test sets as the Ref.[36], and compare the DPA3 model with MACE-OFF23(M) (MACE(M) in short), MACE-OFF23(L) (MACE(L) in short), EScAIP[54] and eSEN[41] models. In this benchmark, we evaluate the DPA3 model with configurations of 6, 12, and 24 layers, denoted as DPA3-L6, DPA3-L12, and DPA3-L24, respectively. Additional hyper-parameters of the model are detailed in Section S-2.2.

To deliver a comprehensive evaluation of a model's performance on a test case consisting of multiple datasets, we introduce an averaged error metric calculated in logarithmic space: the Logarithmic Weighted Average Root Mean Square Error (LWARMSE) or Mean Absolute Error (LWAMAE), defined as follows:

$$\text{LWARMSE or LWAMAE}(\{\mathbf{err}_i, w_i\}) = \exp\left(\frac{1}{\sum_i w_i}\sum_i w_i \log(\mathbf{err}_i)\right), \tag{1}$$

where $\mathbf{err}_i$ denotes the RMSE or MAE (of energy or force) calculated for the i^{th} dataset within the test case, and w_i represents the corresponding weight assigned to each dataset. In the absence of explicitly specified weights, a default value of 1.0 is assigned to each subset.

As depicted in Fig. 2(a) and further detailed in Supplementary Information Table S-1, our smallest model, the DPA3-L6, achieved approximately 34% lower energy LWAMAE compared to MACE(L), despite utilizing only 20% parameters (1.3M vs 6.9M) and delivering a 3.7-fold increase in inference speed. The medium configuration, DPA3-L12, achieved approximately 45% lower energy LWAMAE compared to MACE(L), while achieving an inference speed comparable to MACE(M), representing a favorable trade-off between accuracy and efficiency. Remarkably, the larger DPA3-L24 configuration significantly surpasses MACE(L), achieving a reduction in energy LWAMAE by approximately 66% while utilizing 30% fewer parameters (4.9M vs. 6.9M) and maintaining comparable inference speed. In comparison to EScAIP, DPA3-L24 exhibited superior performance across all energy MAE metrics and five out of seven force MAEs, utilizing only about 11% of EScAIP's parameter count. The eSEN model demonstrated the best performance among all models. The DPA3 family reveals a clear trend: as the model size increases, both energy and force errors decrease, suggesting that its performance may be further enhanced by employing deeper model configurations.

The accuracy of the SPICE-MACE-OFF trained DPA3 model in calculating the torsion profile is benchmarked using the TorsioNet-500[46] test case. As illustrated in Fig. 2(b) and detailed in Table S-2, DPA3-L24 demonstrates superior performance across all metrics compared to models trained with SPICE-MACE-OFF, MACE(M/L), and other models such as AIMNet2[37] and DPA2-drug[47]. Notably, it achieves a 60% reduction in torsional barrier height MAE (MAEB) compared to MACE(L). Remarkably, both DPA3-L12 and DPA3-L24 exhibit zero NABH_h, indicating that the error in all predictions of torsional barrier height falls within the 1 kcal/mol threshold.

The performance of the DPA3 models in condensed-phase systems is preliminarily evaluated using a test case involving liquid water and ice Ih configurations. These configurations were derived from classical ab initio molecular dynamics (AIMD) and path-integral AIMD (PI-AIMD) simulations conducted under various thermodynamic conditions, utilizing the PBE0-TS XC functional[8]. The liquid water configurations was sampled at ambient condition, while the ice Ih configurations was sampled at three thermodynamic states: state (b) 273 K and 1 bar, state (c) 330 K and 1 bar and state (d) 238 K and 2.13 kbar. In accordance with the protocol established in previous research[10], which utilized only 0.1% of the data to demonstrate model data efficiency, we adopted the same settings. This approach facilitates direct comparison among the DeepPot-SE[55], NequIP[10], and DPA3 models.

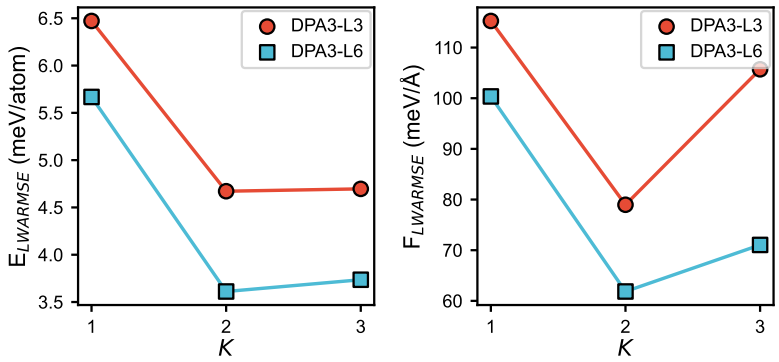

Figure 3. LWARMSEs in energy and force predictions evaluated on the DPA2 test sets. The DPA3 models with three layers (DPA3-L3) and six layers (DPA3-L6) were examined at varying LiGS orders K.

As shown in Fig. 2(c) and detailed in Table S-3, DPA3-L12 achieves overall lower RMSE values compared to NequIP, with around 60% reduction in energy LWARMSE and 30% reduction in force LWARMSE. Notably, it even outperforms the original DeePMD model, which was trained on the complete dataset, in one out of four energy RMSEs and three out of four force RMSEs, underscoring DPA3's strong fitting capability. Interestingly, the scaling law is less apparent in this context: DPA3-L12 exhibits superior overall performance compared to DPA3-L24. This implies that the diversity of configurations within the test case is limited, causing larger models to be prone to overfitting. Furthermore, our evaluation of DeepPot-SE—a refined version of the original DeePMD—under identical training conditions demonstrated comparable performance to NequIP, despite utilizing only 0.1% of the data. This finding challenges the validity of data efficiency assessments for this specific dataset[10], emphasizing the necessity for more representative benchmarks to systematically evaluate model performance.

DPA3 was further evaluated using a test case proposed in Ref.[48], consisting of three datasets: formate decomposition on Cu, defected bilayer graphene, and zeolites. These datasets exemplify catalysis, two-dimensional materials, and porous materials, respectively. As shown in Fig. 2(d) and Table S-4, the smallest model, DPA3-L6, consistently outperforms NequIP and demonstrates lower errors than AlphaNet, except for force prediction in the graphene dataset. Improved accuracy is achieved with larger models, DPA3-L12 and DPA3-L24, both of which consistently outperform NequIP and AlphaNet on the first two datasets. For the zeolite dataset, a clear trend of reduced error rates with larger DPA3 architectures is observed. The DPA3-L24 model demonstrates only slightly lower accuracy compared to AlphaNet.

The DPA3 model is benchmarked using the DPA2 test sets, which were originally proposed for evaluating the DPA2 model in Ref.[24]. These test sets comprise 18 datasets spanning diverse research domains, including alloy and battery materials, metal cluster catalysis, drug-like molecules, and linear alkane pyrolysis, all trained under an identical protocol. The LWARMSEs, calculated using the weights outlined in Table S-5, are depicted in Fig. 2(e). According to this metric, DPA3-L24 consistently demonstrates superior performance in both energy and force predictions, reinforcing its reliability across various applications. Notably, EqV2 shows a lower force RMSE in certain systems, as detailed in Table S-5. This may be attributed to its non-conservative force-prediction approach, which fits energy and force separately, at the expense of conservativeness.

The performance of the DPA3 architecture, constructed with varying orders of the LiGS, is evaluated using the DPA2 test sets. Specifically, we assessed the LWARMSEs of both energy and forces for DPA3-L3 and DPA3-L6 under different LiGS orders K, as illustrated in Fig. 3. The model's accuracy improves significantly when increasing the order from 1 to 2; however, this trend does not persist as the order

Table 1. Results on the Matbench Discovery leaderboard, with all compliant models accessed before May 27, 2025.

Model	CPS↑	Acc↑	F1↑	DAF↑	Prec↑	MAE↓	R2↑	κSRME↓	RMSD↓	Params↓	Targets
DPA3-L24	0.717	0.936	0.803	5.024	0.768	0.037	0.812	0.650	0.080	4.92M	EFSG
eSEN-30M-MP	0.797	0.946	0.831	5.260	0.804	0.033	0.822	0.340	0.075	30.1M	EFSG
SevenNet-l3i5	0.714	0.920	0.760	4.629	0.708	0.044	0.776	0.550	0.085	1.17M	EFSG
MatRIS v0.5.0 MPtrj	0.681	0.938	0.809	5.049	0.772	0.037	0.803	0.861	0.077	5.83M	EFSGM
GRACE-2L-MPtrj	0.681	0.896	0.691	4.163	0.636	0.052	0.741	0.525	0.090	15.3M	EFSG
MACE-MP-0	0.644	0.878	0.669	3.777	0.577	0.057	0.697	0.647	0.091	4.69M	EFSG
AlphaNet-MPTrj	0.566	0.933	0.799	4.863	0.743	0.041	0.745	1.310	0.107	16.2M	EFSG
eqV2 S DeNS	0.522	0.941	0.815	5.042	0.771	0.036	0.788	1.676	0.076	31.2M	EFSD
ORB v2 MPtrj	0.470	0.922	0.765	4.702	0.719	0.045	0.756	1.725	0.101	25.2M	EFSD
M3GNet	0.428	0.813	0.569	2.882	0.441	0.075	0.585	1.412	0.112	228k	EFSG
CHGNet	0.400	0.851	0.613	3.361	0.514	0.063	0.689	1.717	0.095	413k	EFSGM

increases to 3. In fact, a decrease in accuracy is observed when increasing the order from 2 to 3, particularly in the force accuracy. In theory, the order-2 DPA3 possesses a higher capacity than the order-3, which suggests the observed decline may be attributed to two factors: First, the edge features in the order-2 graph, geometrically represented as angular features, sufficiently capture the local geometric context for most systems. Second, the inclusion of edge features in the order-3 graph, specifically dihedral terms, potentially complicates the training process. Based on these findings, we adopt order $K = 2$ as the default configuration, as it provides the optimal performance.

2.3 Scaling Law

The performance of DPA3 is validated on well-established Matbench Discovery leaderboard[56], a benchmark that ranks MLIPs across various tasks simulating the high-throughput discovery of stable inorganic crystals. Specifically, compliant models trained on the MPtrj dataset[17] are evaluated on the WBM test set[57] by performing structural optimization and predicting formation energies, subsequently converted to convex hull distances to assess thermodynamic stability. Table 1 summarizes the comparative results between DPA3-L24 and other state-of-the-art compliant models accessed from the leaderboard on May 27, 2025. DPA3 achieves the second-best combined performance score (CPS), demonstrating an excellent balance between model complexity (parameter count) and predictive performance.

The scaling law of the DPA3 model is assessed using the MPtrj dataset, with generalization error evaluated on a randomly down-sampled WBM dataset[57], consisting of 25,000 configurations, approximately one-tenth of the original test set. This evaluation involves a multifactorial scaling analysis during training, with a focus on the model scale M, data scale D, and compute budget C. For the DPA3 model, the model scale is modified by varying the number of layers, denoted by L, from 3 to 24, while ensuring that the number of parameters per layer remains consistent. The batch size B is kept constant across different scales, enabling training steps S to serve as a proxy for D ($D \approx B \times S$). The compute budget C is approximated as $M \times S$, thus simplifying the intricate relationship between computational costs and $M \times S$. In accordance with Ref.[31], we employ the following empirical power-law relationship:

$$\mathrm{MAE}(M,S,C) = \alpha_m M^{\beta_m} + \alpha_s S^{\beta_s} + \alpha_c C^{\beta_c}. \tag{2}$$

The parameters α_m, α_s, α_c, β_m, β_s, and β_c are fitted using five independent training runs of the DPA3 model with different numbers of layers, with generalization MAE evaluated every 100,000 training steps

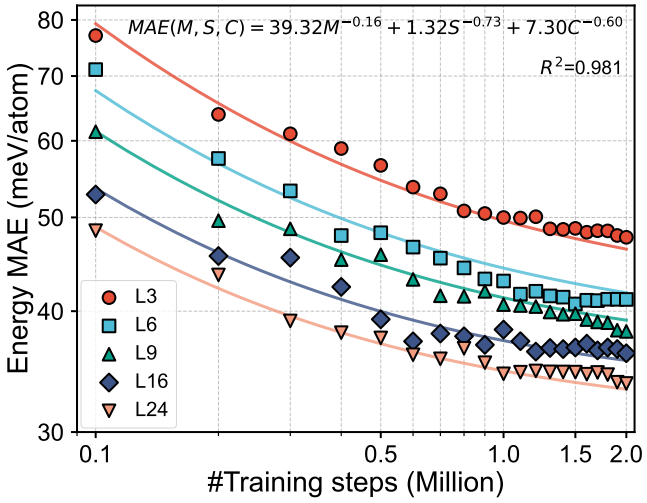

Figure 4. Scaling law of the DPA3 models. All evaluations are conducted by measuring test energy MAEs on the subsampled WBM dataset using models trained on the MPtrj dataset. DPA3 exhibits smooth performance improvement with jointly scaling of model parameters (M), training steps (S), and compute budget (C).

(other hyperparameters are detailed in Table. S-6). Consequently, we have:

$$\alpha_m = 39.32, \quad \alpha_s = 1.32, \quad \alpha_c = 7.30 \tag{3}$$

$$\beta_m = -0.16, \quad \beta_s = -0.73, \quad \beta_c = -0.60 \tag{4}$$

As illustrated in Figure 4, D, M, and C exhibit direct, quantifiable impacts on test MAEs. The high R^2 value of 0.981 signifies a robust correlation between our scaling law and the observed MAE data, validating the DPA3 model's scaling law and demonstrating its potential for improved performance through systematic scaling.

2.4 DPA-3.1-3M

The DPA3 architecture is utilized to train a LAM using the OpenLAM-v1 collection, which comprises 31 datasets, as described in Refs.[26,42]. The OpenLAM-v1 encompasses a diverse range of research domains, including organic materials, organic and bio-molecules, catalysis, and chemical reactions. Notably, this collection incorporates large-scale open-source datasets like OMat24[33], OC20[58], SPICE2[38], and those used to train DPA2[24]. Due to the inconsistent DFT settings across these datasets, we utilize a multi-task training scheme to concurrently train the model across all datasets[23,24]. The resultant model, termed DPA-3.1-3M, is truncated at order $K = 2$ and comprises $L = 16$ layers. Under these specifications, the model encompasses approximately 3.26 million parameters.

The generalizability of the DPA-3.1-3M model is evaluated through its zero-shot prediction accuracy in the force field tasks of the LAMBench benchmark suite[26]. In the benchmark, the model's accuracy is assessed across 17 downstream datasets that span a wide range of research domains. These datasets were proposed in the literature as training resources for MLIPs designed to address various scientific challenges. Importantly, they were independently constructed from the training datasets within the OpenLAM-v1 collection. In this context, zero-shot prediction refers to the direct evaluation of the DPA-3.1-3M model's prediction error on the downstream datasets without additional training. It is noteworthy that the DPA-3.1-3M model is employed to predict the energy difference between the labels and a dummy model that

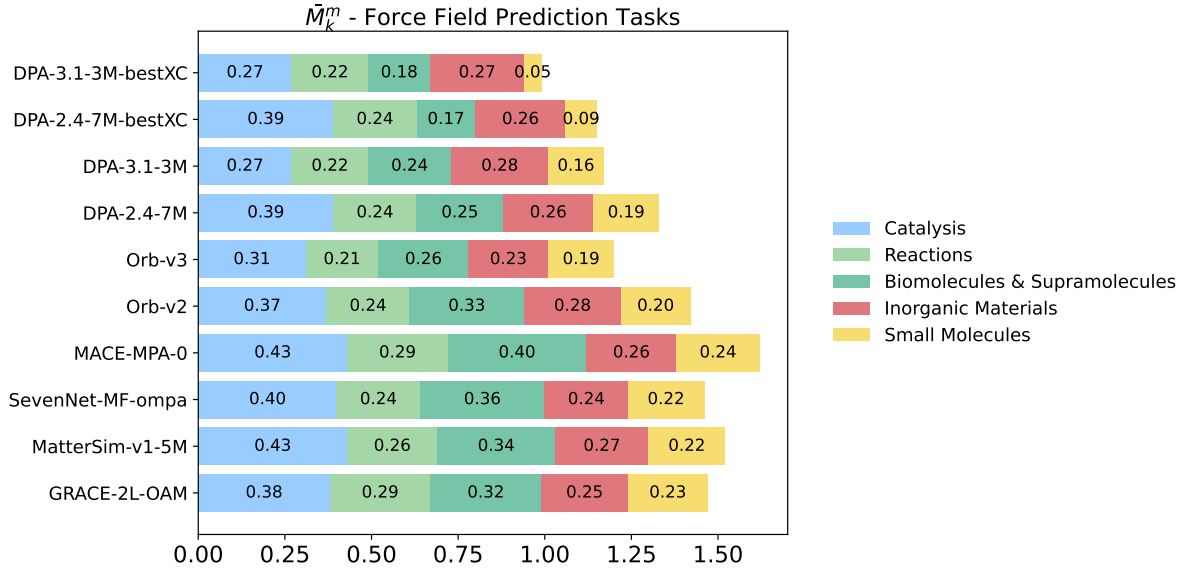

Figure 5. Generalizability evaluation in LAMBench-v0.2.1[26] with dimensionless error metrics for force field tasks across five distinct domains. Each domain consists of one or more independently constructed datasets from the literature, each designed to train MLIPs for specific scientific challenges, thereby making them out-of-distribution relative to the training data. The evaluation included a total of 17 datasets.

linearly correlates the DFT energy with chemical composition. The accuracy of the DPA-3.1-3M model, along with other state-of-the-art LAMs released prior to May 1, 2025, is evaluated using the dimensionless error metric introduced by the LAMBench[26]. This metric integrates energy, force, and, where applicable, virial errors across datasets from various research domains into a single dimensionless value. A value of 0 indicates perfect alignment with DFT predictions, while a value of 1 signifies an error equivalent to that of a dummy model. A lower error metric implies that the model would either provide superior accuracy as a standalone PES model or require less data for fine-tuning in downstream applications.

The predictions produced by the multi-task trained DPA-3.1-3M model are influenced not only by the atomic coordinates and types within the system but also by the choice of dataset encoding. The ideal dataset encoding is one whose XC functional closely aligns with the downstream test case and shares similar chemical and configurational spaces. For our evaluation, we primarily utilized the MPtrj dataset code for most test tasks. However, for the ANI-1x dataset (labeled with ωB97X/6-31G*) and the AIMD-Chig dataset (labeled with M06-2X/6-31G*), we employed the Yang2023ab[47] dataset code (labeled with ωB97X-D/6-31G**). Furthermore, when testing the Sours2023Applications[59] dataset, the D3(BJ) dispersion correction was applied to enhance predictions with the MPtrj code. The error metric for these conditions is denoted by the label *DPA-3.1-3M-bestXC* in Fig. 5. Additionally, the performance of the DPA-3.1-3M model was evaluated across all test cases with the MPtrj code, as labeled by *DPA-3.1-3M* in Fig. 5, without any presumption of *a priori* knowledge regarding the training or downstream XC functional.

As shown in Figure 5, the generalizability of the DPA-3.1-3M model is compared with contemporary LAMs including DPA-2.4-7M[24,26] (also DPA-2.4-7M-bestXC, utilizing identical data code selection to DPA-3.1-3M), Orb-v2[60]/v3[61], MACE-MPA-0[19], SevenNet-MF-ompa[62], MatterSim-v1-5M[20] and GRACE-2L-OAM[63]. Without explicitly considering data code selection, the multi-task pretrained DPA-3.1-3M demonstrates overall state-of-the-art performance, reaffirming both the effectiveness of multi-task pretrain-

ing and the robustness of the DPA3 architecture in LAM implementations. Notably, DPA-3.1-3M-bestXC further enhances performance, particularly excelling in the Small Molecules and Reactions domains, underscoring the effectiveness of data encoding and significance of prior knowledge regarding XC functionals in improving accuracy. Additionally, Orb-v3 demonstrates commendable performance, achieving the best results in the Reactions and Inorganic Materials domains. The benchmark results once again underscore the critical importance of multi-task training. As the number of datasets continues to grow, adopting a multi-task training strategy becomes essential for developing LAMs that can generalize effectively across diverse domains and thereby tackle real-world scientific challenges.

3 Discussion

In this work, we introduce the DPA3 model architecture, specifically designed to meet the requirements of large atomistic models (LAMs). The DPA3 model adheres to the physical principles inherent in universal potential energy surfaces (PES), exhibiting scaling laws and maintaining a constant parameter scale that remains independent of the number of tasks in a multi-task training scheme. The performance of the DPA3 model is demonstrated through two benchmarking approaches. Firstly, the DPA3 model is assessed in standard potential energy fitting tasks that are well-established in the literature. Its accuracy is assessed across five benchmark cases spanning diverse research domains, including molecular systems, bulk materials, surface and cluster catalysis, two-dimensional materials, and battery materials. The model demonstrates superior performance in the majority of test cases. Secondly, the model is evaluated as an LAM, specifically the DPA-3.1-3M model trained on the OpenLAM-v1 dataset under a multi-task scheme, using the LAMBench benchmark suite. The DPA-3.1-3M model showcases exceptional generalizability in zero-shot force field tasks within LAMBench, indicating its ability to achieve high accuracy as an out-of-the-box potential energy model for scientific inquiries, or requiring minimal data when fine-tuned for downstream tasks. The DPA3 model is characterized by its smoothness and conservativeness, indicating that higher accuracy in force field generalizability typically correlates with enhanced performance in property calculation tasks[41].

The pathway to enhancing the generalizability of the DPA3 model as an LAM is evident. According to the scaling law, increasing the model size can be achieved through optimizing the implementation or by integrating model parallelization mechanisms. Additionally, expanding the scale of training data by incorporating datasets from a diverse range of application domains will further enhance the model's generalizability. For instance, integrating the recently published OMol25 dataset[64], which comprises 102 million configurations, into the training of the DPA3 model is expected to remarkably enhance its generalizability, especially within molecular systems.

It is noteworthy that the DPA3 model currently employs invariant features within the LiGS framework. Its performance is observed to be suboptimal compared to equivariant models such as MACE when tested on datasets generated by AIMD simulations of small molecules, including the rMD17 dataset[65], 3BPA dataset[66] and acetylacetone dataset[67]. The potential advantages of incorporating equivariant features to enhance generalizability in these molecular systems remain uncertain and warrant further investigation.

4 Methods

4.1 Datasets

Here, we elaborate on all the benchmark datasets used in Section 2.2, which span both small molecular systems (e.g. SPICE-MACE-OFF) and material systems (e.g. Water, Zeolite, MPtrj, etc.).

The SPICE-MACE-OFF[36]. This small-molecule dataset, developed for training the MACE-OFF23 model[36], combines 85% of the SPICE[68] v1 subset (neutral molecules containing H, C, N, O, F, P, S, Cl, Br, and I) with geometries from classical MD simulations at 300K and 500K for diverse conformations ($\leq$ 50 atoms). It extends to larger systems through GFN2-xTB[69] MD-generated 50–90 atom molecules (QMugs-derived[70]) and water clusters (up to 50 molecules) from liquid water MD trajectories. All energies and forces were computed at the ωB97M-D3(BJ)/def2-TZVPPD[49–53] level using PSI4[71]. Data partitioning follows molecule-level splitting, 95% training/validation (951,005 structures) and 5% testing (50,195 structures), to prevent conformation overlap. Additional test cases included COMP6 tripeptides[72] recomputed at the SPICE level of theory; however, due to the unavailability of these tripeptides in the test set, accuracy comparisons were limited to other molecular systems.

The TorsionNet-500[46]. This dataset comprises 500 drug-like molecules (H, C, N, O, F, S, Cl) with 24 conformations per molecule, generated by rotating a single bond in 15° increments. These structures were initially optimized at the B3LYP/6-31G(d) DFT level and span diverse pharmaceutical chemical space.

Liquid Water and Ice Dynamics[8]. This dataset consists of reference structures for training and validating machine learning potentials (MLPs), encompassing liquid water and ice Ih configurations generated from classical AIMD and PI-AIMD simulations under varying thermodynamic conditions (1-2.13 kbar, 238-330 K) using the PBE0-TS functional. It contains 140,000 structures partitioned into 100,000 liquid water configurations (64 H_2O molecules) and 40,000 ice Ih configurations (96 H_2O molecules), including three distinct ice phases simulated at PI-AIMD and classical AIMD levels with different pressure-temperature combinations. We uniformly sampled <0.1% of the data (133 structures) for training consistent with previous work[10], with 50 frames allocated for validation and all remaining data reserved for testing.

The Formate Decomposition on Cu[10]. This dataset consists of configurations characterizing the decomposition process of formate on Cu(110), focusing on C-H bond cleavage. It includes initial states (monodentate/bidentate formate), intermediate configurations, and final states (H ad-atom with gas-phase CO_2). The Nudged Elastic Band (NEB) method generated reaction pathways, followed by 12 *ab initio* molecular dynamics (AIMD) simulations using the CP2K[73] code. These simulations produced 6,855 DFT structures with a 0.5 fs time step over 500-step trajectories, capturing dynamic evolution across reaction coordinates. The dataset provides atomistic-scale insights into catalytic decomposition mechanisms through systematically sampled configurations. The full dataset was partitioned into training (2,500 structures), validation (250 structures), and test (remaining 4,105 structures) sets via uniform random sampling.

The Defected Bilayer Graphene[74]. This dataset consists of reference structures designed to train and validate MLPs, encompassing three bilayer systems: V0V0 (pristine), V0V1 (single vacancy on the top layer), and V1V1 (single vacancy in both layers). The dataset includes single-point DFT (PBE+MBD) energies and atomic forces for configurations with varying interlayer distances, stacking modes, and manual deformations, supplemented by snapshot configurations from classical and DFT-based molecular dynamics (MD) simulations at different temperatures. The data were partitioned into training (3,988 structures), validation (4,467 structures), and test (200 structures) sets. The splitting was based on farthest point sampling (FPS), by performing principal components analysis (PCA) and choosing sufficiently distant random points for different sets to ensure representative sampling.

The Zeolite Dataset[48]. This dataset comprises 16 different types of zeolite relevant to catalysis, adsorption, and separation applications. It contains atomic trajectories generated through AIMD simulations at 2000 K using VASP[75,76], with 80,000 snapshots per zeolite providing calculated energies and atomic

forces. The dataset is partitioned into training (48,000 structures), validation (16,000), and test (16,000) sets using a random 6:2:2 split ratio. This standardized division ensures systematic evaluation of MLPs while maintaining consistency across computational frameworks.

The DPA2 Test Sets[24]. This composite dataset comprises 18 specialized sub-datasets (totaling 5,119,379 structures: 4,045,094 training, 1,074,285 test) for pre-training the DPA2 model, integrating **domain-specific collections** including metallic alloys (Alloy), cathode materials (Cathode-P), semiconductors (SemiCond-P), drug-like molecules (Drug), catalytic reaction trajectories (OC2M) and etc., alongside with specialized systems such as H_2O configurations, metallic materials (Sn, AgAu, AlMgCu), and n-dodecane pyrolysis (C12H26). The '-P' suffix indicates datasets reformulated with pretraining splits in the DPA2 paper (see Ref.[24] for details). Aggregated from the DeepModeling community[77] and external sources, these 73-element datasets employ calculations from various DFT softwares like the VASP[75,76], Gaussian[78], and ABACUS[79,80] for multi-task training. Following Ref.[24], we conduct per-dataset training with averaged error metrics for fair benchmarking.

Materials Project Trajectory Dataset (MPtrj)[17]. This dataset provides a comprehensive collection of DFT calculations for ∼146,000 inorganic materials spanning 89 elements, derived from 1.37 million structural relaxation and static calculation tasks in the Materials Project Database[81]. Calculations employ generalized gradient approximation (GGA) or GGA+U exchange-correlation methods, generating 1.58 million atomic configurations with associated energies, magnetic moments (7.94 million), atomic forces (49.3 million), and stress tensors (14.22 million). By systematically sampling trajectories from relaxation pathways and equilibrium states, the dataset captures the potential energy surfaces from diverse regions of inorganic materials, serving as a foundational resource for training and benchmarking machine learning models in computational materials science. We employ this dataset to train models and test their ability to predict ground-state (0 K) thermodynamic stability through geometry optimization and energy prediction within the Matbench Discovery Benchmark [56].

4.2 DPA3 model architecture

In this study, we investigate a system composed of N atoms, where the atomic numbers are represented by the list $\mathscr{Z} = (Z_1, \ldots, Z_i, \ldots, Z_N)$ and the atomic positions are denoted by $\mathscr{R} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_i, \ldots, \boldsymbol{r}_N)$. The PES of the system, indicated by $E = E(\mathscr{Z}, \mathscr{R})$, is postulated to be decomposable into a sum of atomic contributions E_i, such that $E = \sum_i E_i$. The atomic force F_i exerted on atom i and the virial tensor Ξ_{pq} are defined as the negative gradients of the total energy with respect to the atomic coordinates and the simulation cell vectors, respectively:

$$F_i = -\nabla_{\boldsymbol{r}_i} E, \quad \Xi_{pq} = -\sum_r \frac{\partial E}{\partial h_{rp}} h_{rq}. \tag{5}$$

In Eq. (5) Ξ_{pq} corresponds to the pq component of the virial tensor, and h_{pq} yields the q-th component of the p-th cell vector.

The DPA3 model constructs the atomic energy contribution as follows:

$$E_i = \mathscr{F}\left(v_i^{(1,L)}, c(\mathscr{D}_m)\right) + e_m(Z_i), \tag{6}$$

In this formulation, $\mathscr{F}$ denotes the fitting network, with $v_i^{(1,L)}$ representing the vertex feature of graph $G^{(1)}$ after L layers of updates. This vertex feature functions as a descriptor of the atomic environment surrounding atom i. Furthermore, $c(\mathscr{D}_m)$ refers to the encoding of the dataset $\mathscr{D}_m$. The energy bias, $e_m(Z_i)$,

is obtained through least squares fitting to the training energy of dataset $\mathscr{D}_m$, utilizing the chemical formula. This approach effectively mitigates the arbitrariness associated with selecting the energy zero point across different DFT calculations.

The DPA3 model evolves vertex and edge features through a recursive formulation applied to each graph within the LiGS. Let $v_\alpha^{(k,l)}$ and $e_{\alpha\beta}^{(k,l)}$ denote the vertex and edge features of graph k after l layers, respectively. The features of the $l+1$ layer are initialized by the features of the previous layer, i.e., $v_\alpha^{(k,l+1)} \leftarrow v_\alpha^{(k,l)}$ and $e_{\alpha\beta}^{(k,l+1)} \leftarrow e_{\alpha\beta}^{(k,l)}$. The features are subsequently refined using a residual update mechanism:

$$v_\alpha^{(k,l+1)} \leftarrow v_\alpha^{(k,l+1)} + \delta_c^{(k,l)} u_\alpha^{(k,l)}, \tag{7}$$

$$e_{\alpha\beta}^{(k,l+1)} \leftarrow e_{\alpha\beta}^{(k,l+1)} + \delta_s^{(k,l)} m_{s,\alpha\beta}^{(k,l)}, \tag{8}$$

where $\delta_c^{(k,l)}$ and $\delta_s^{(k,l)}$ represent trainable step sizes for the updates. The term $u_\alpha^{(k,l)}$ indicates the vertex feature update, defined as a convolution of messages of all the edges connecting to the vertex. $m_{s,\alpha\beta}^{(k,l)}$ denotes the self-message used in the edge feature update. These updates are defined as follows:

$$u_\alpha^{(k,l)} = \phi_u \left(N_m^{-\alpha_k} \sum_{\beta \in \mathscr{E}^{(k)}(\alpha)} w_{\alpha\beta}^k m_{c,\alpha\beta}^{(k,l)} \right), \tag{9}$$

$$m_{c,\alpha\beta}^{(k,l)} = \phi_c(v_\alpha^{(k,l)}, v_\beta^{(k,l)}, e_{\alpha\beta}^{(k,l)}) \tag{10}$$

$$m_{s,\alpha\beta}^{(k,l)} = \phi_s(v_\alpha^{(k,l)}, v_\beta^{(k,l)}, e_{\alpha\beta}^{(k,l)}) \tag{11}$$

where ϕ_u may be either a multi-layer perceptron (MLP) or an identity mapping, and ϕ_c and ϕ_s are MLPs. In Eq. (9), $\mathscr{E}^{(k)}(\alpha)$ represents the set of all edges connected to vertex α in the graph $G^{(k)}$. The normalization factor $N_m^{-\alpha_k}$ involves N_m, which estimates the maximum possible number of neighbors, and a power factor $\alpha_k > 0$. Notably, the vertex feature $v^{(k,l)}$ is identical to the edge feature $e^{(k-1,l)}$ from the preceding graph in the LiGS. Consequently, we avoid maintaining redundant information regarding vertex features for any graph $G^{(k)}$ where $k > 1$. Instead, the vertex feature updates $u_\alpha^{(k,l)}$ are transferred to graph $G^{(k-1)}$ to update the edge features $e^{(k-1,l)}$. Subsequently, the updated edge features $e^{(k-1,l+1)}$ of graph $G^{(k-1)}$ determine the vertex feature $v^{(k,l+1)}$ of graph $G^{(k)}$, as depicted in Fig. 1(e).

In Eq. (9), the message $m_{c,\alpha\beta}^{(k,l)}$ is appropriately smoothed by incorporating a prefactor $w_{\alpha\beta}^k$ to ensure that it gradually diminish as the distance between the atoms approaches the cut-off radius at different orders k. Specifically, in $G^{(1)}$, we define a smooth switch function $s^k(r_{ij})$ that exponentially decays from 1 to 0 as the distance r_{ij} defined increases from 0 to a cutoff radius r_c^k. The switch function is defined as:

$$s^k\left(r_{ij}\right) = \begin{cases} \exp(-\exp(C(r_{ij} - r_{cs}^k)/r_{cs}^k)) & \text{if } 0 < r_{ij} \le r_c^k, \\ 0 & \text{if } r_{ij} > r_c^k, \end{cases} \tag{12}$$

where r_{cs}^k is a manually adjustable smoothing factor and C is a tunable hyper-parameter controlling the decay rate. Typically, for $k = 1$, we set $w_{ij}^1 = s^1(r_{ij})$, with $C = 20$, $r_c^1 = 6.0$Å, and $r_{cs}^1 = 5.3$Å. For higher-order terms ($k > 1$), a similar approach is used, but the prefactor $w_{\alpha\beta}^k$ is defined as the product of multiple switch functions, each acting on the distances defined within $G^{(1)}$, ensuring overall smooth

diminishing. In graph $G^{(2)}$, the edge $\alpha\beta$ is represented by the angle formed between two atomic bonds that share a common vertex, specifically $(ij)(im)$. Thus,

$$w^2_{\alpha\beta} = w^2_{(ij)(im)} = s^2(r_{ij}) \times s^2(r_{im}). \tag{13}$$

In graph $G^{(3)}$, the edge $\alpha\beta$ is represented by the dihedral formed between two angles that share a common bond, specifically $(ijm)(ijn)$.

$$w^3_{\alpha\beta} = w^3_{(ijm)(ijn)} = s^3(r_{ij}) \times s^3(r_{im}) \times s^3(r_{in}). \tag{14}$$

For the initial graph $G^{(1)}$, vertex features are iteratively refined by incorporating additional self-message and symmetrization transformation terms as follows (see Fig. 1 (d)):

$$v_i^{(1,l+1)} \leftarrow v_i^{(1,l+1)} + \delta_{s,0}^{(1,l)} \phi_{s,0}(v_i^{(1,l)}) + \delta_{s,1}^{(1,l)} \phi_{s,1}(\tilde{v}_i^{(1,l)}). \tag{15}$$

In this equation, $\phi_{s,0}$ and $\phi_{s,1}$ are the MLPs responsible for processing the self-message and the symmetrization-transformed intermediate features $\tilde{v}_i^{(1,l)}$, respectively. The intermediate representation $\tilde{v}_i^{(1,l)}$ is defined by

$$\tilde{v}_i^{(1,l)} = \text{concat}\left(\text{symm}\left(v_j^{(1,l)}, h_{ij}\right), \text{symm}\left(e_{ij}^{(1,l)}, h_{ij}\right) \right). \tag{16}$$

Here, h_{ij} is a three-component vector given by $h_{ij} = \frac{s^1(r_{ij})}{(r_{ij})^2} \times (x_{ij}, y_{ij}, z_{ij})$, where (x_{ij}, y_{ij}, z_{ij}) represent the Cartesian components of the vector $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$, $r_{ij} = |\boldsymbol{r}_{ij}|$ and $s^1(r_{ij})$ is the smooth switch function defined in Eq. (12) with $k = 1$. The symm operator, as introduced by Zhang et al. (2018), is defined generally as $\text{symm}(x_j, y_j)$, with x_j and y_j being neighbor-indexed vectors. It is assumed that x_j is rotationally invariant, whereas y_j is not, although their inner product maintains rotational invariance. The symmetrization operator is mathematically represented by:

$$\text{symm}(x_j, y_j) = \underset{ps}{\text{flatten}}\left(\sum_q z_{pq} z^<_{sq} \right), \tag{17}$$

$$z_{pq} = (N_m^{-1}) \sum_{j \in N_{r_c^1}(i)} w^1_{ij} x_{j,p} y_{j,q}, \tag{18}$$

$$z^<_{pq} = \underset{p}{\text{split}}(z_{pq}). \tag{19}$$

In Equation (17), the matrix dimensions p and r are flattened into a vector. Equation (18) involves a summation over neighboring indices j, ensuring that the matrix z remains permutationally invariant. Equation (19) describes the splitting of the matrix z_{pq} along the dimension p, where the initial subset of elements is denoted by $p^<$. It can be demonstrated that the symmetrization operator is invariant under rotational operations and permutations of atoms sharing the same atomic number, as detailed by Ref.[55].

The vertex features of graph $G^{(1)}$ correspond to attributes associated with the atoms within the system and are initialized as follows:

$$v_\alpha^{(1,0)} = v_i^{(1,0)} = \text{one\_hot}(Z_i), \tag{20}$$

where one_hot denotes the one-hot encoding of atomic number Z_i. The edge features of the graph represent neighboring atom pairs and are initialized through:

$$e_{\alpha\beta}^{(1,0)} = e_{ij}^{(1,0)} = \phi_e^{(1)}(r_{ij}), \tag{21}$$

where $\phi_e^{(1)}$ is an MLP designed to embed the interatomic distances between pairs of atoms. In graph $G^{(2)}$, the edge $\alpha\beta$ corresponds to an angle formed by two bonds (ij) and (im) sharing a common vertex. The edge features $e_{\alpha\beta}^{(2,0)}$ are initialized using:

$$e_{\alpha\beta}^{(2,0)} = e_{(ij)(im)}^{(2,0)} = \phi_e^{(2)}(\cos(\theta_{ijm})), \tag{22}$$

where $\phi_e^{(2)}$ represents the embedding network, and θ_{ijm} is the angle formed by the atom pairs ij and im. For graph $G^{(3)}$, the edge $\alpha\beta$ corresponds to a dihedral formed by two angles (ijm) and (ijn) sharing a common bond (ij). The edge features are initialized by:

$$e_{\alpha\beta}^{(3,0)} = e_{(ijm)(ijn)}^{(3,0)} = \phi_e^{(3)}(\cos(\eta_{mijn})), \tag{23}$$

where $\phi_e^{(3)}$ denotes the embedding network, and η_{mijn} is the dihedral angle involving the angle objects jim and jin. These initializations ensure that each graph layer is equipped with features pertinent to its respective structural relations.

To demonstrate the invariance of the DPA3 model under translation, rotation, and permutation operations, it is sufficient to establish that the vertex feature of the final layer, $v_i^{(1,L)}$, remains equivariant under permutation operations and invariant under translational and rotational operations.

For translational and rotational symmetries, it is sufficient to confirm that both feature initializations and layer-wise updates maintain these symmetries. The vertex and edge features of the graphs are initialized according to Eqs. (20)–(23), which consider atomic numbers, interatomic distances, angles, and dihedrals - all of which are invariant under translational and rotational transformations. Moreover, since the inputs to the messages in Eqs. (10) and (11) are translationally and rotationally invariant, the messages themselves are also invariant under these operations. Consequently, the layer-wise feature updates preserve translational and rotational invariance.

To examine the equivariance concerning permutational operations over atoms of identical chemical species, we initially note that the vertex set, edge set, and connections remain invariant under these operations across all graphs in the LiGS. Next, we consider feature initialization. The initial vertex features in graph $G^{(1)}$, as described by Eq. (20), are equivariant. Although the order of the edge features $e_{\alpha\beta}^{(k,0)}$ for any $k \geq 1$ may change, these altered features can be mapped back to the original set via a permutation. During the feature update process, the messages defined in Eqs. (10) and (11) exhibit the same permutation pattern as the edges. Given that the connections $\mathscr{E}^{(k)}$ within any graph k remain unchanged, the convolution in Eq. (9) is aligned with the vertex permutation. This alignment is attributable to the fact that the order of neighbors β is irrelevant due to the summation involved. Consequently, according to Eqs. (7) and (8), both vertex and edge updates maintain the same permutation pattern as their respective features. Since the vertex features of the graph $G^{(1)}$ are permuted in the same manner as the atomic permutation, the permutational equivariance of the final layer $v_i^{(1,L)}$ is confirmed.

5 Acknowledgments

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). The work of Jinzhe Zeng was supported by the advanced computing resources provided by the Supercomputing Center of the USTC. The work of Tong Zhu is supported by the National Natural Science Foundation of China (Grants No. 22222303 and No. 22173032).

6 Data availability

The OpenLAM-v1 dataset used for training DPA-3.1-3M models is available on AIS Square (https://www.aissquare.com/datasets?search=OpenLAM). The DPA3 codes are available in the DeePMD-kit repository (https://github.com/deepmodeling/deepmd-kit) after version 3.1.0.

Supplementary Information

S-1 Benchmark results

Here we present detailed benchmark results of the DPA3 model across various datasets as discussed in the main text, including SPICE-MACE-OFF (Table S-1), TorsionNet-500 (Table S-2), liquid water and three ice systems (Table S-3), three material datasets from Ref.[48] (Table S-4) and DPA2 test sets (Table S-5).

S-2 Model training

S-2.1 Robustness of activation function

During the training of the DPA3 model, we found that certain normalization techniques (such as layer normalization or batch normalization) had a **negative impact** on both accuracy and efficiency. Consequently, explicit normalization was **not** incorporated into the model architecture.

On the other hand, we employed the SiLU activation function, which outperformed tanh in terms of accuracy; however, it is an unbounded activation function. We observed that during training, particularly with deeper networks, there is a tendency for numerical instability leading to explosive accumulation of values. Therefore, we replaced SiLU with our new activation function, termed SiLUT (SiLU threshold with Tanh), defined as follows:

$$\mathrm{SiLUT}(x) = \begin{cases} \mathrm{SiLU}(x) & \text{if } x \leq t, \\ \tanh(a \cdot (x-t)) + b & \text{if } x > t, \end{cases} \tag{24}$$

where t represents the threshold, indicating when to transition from SiLU to tanh. Constants a and b are determined based on t to ensure the first and second continuity of the activation function at the threshold.

We found that this activation function design, in most cases, maintains the accuracy of SiLU while providing more stable training, especially in the absence of explicit normalization.

S-2.2 Model hyperparameters

The hyperparameters utilized for training the DPA3 models are summarized in Table S-6. All models were trained using a consistent architecture and fixed dimensions for vertex and edge features across different graph orders. To ensure optimal utilization and balancing across different devices, especially using multiple GPUs, batch sizes were dynamically determined based on the number of atoms present in each specific system. An exponential decay strategy was employed for learning rate scheduling throughout the training process, with simultaneous adjustments to the prefactors of loss components in synchronization with these learning rate changes. For both the SPICE-MACE-OFF and MPtrj datasets, training was conducted in two rounds, where the second round was initialized from the checkpoint of the first. Each round employed distinct configurations of loss functions and prefactors, leading to enhanced overall performance.

Table S-1. Test MAE on SPICE-MACE-OFF dataset. Energy (E) MAE is in **meV/atom**, force (F) MAE is in **meV/Å**, and efficiency is in **million steps/day**.

	MACE(M)		MACE(L)		EScAIP[a]		eSEN		DPA3-L6		DPA3-L12		DPA3-L24	
Dataset	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**
PubChem	0.91	20.57	0.88	14.75	0.53	5.86	**0.15**	**4.21**	0.51	15.20	0.41	12.12	0.24	8.47
DES370K M.	0.63	9.36	0.59	6.58	0.41	3.48	**0.13**	**1.24**	0.35	6.37	0.31	5.25	0.18	3.15
DES370K D.	0.58	9.02	0.54	6.62	0.38	2.18	**0.15**	**2.12**	0.46	6.14	0.43	5.13	0.23	3.19
Dipeptides	0.52	14.27	0.42	10.19	0.31	5.21	**0.07**	**2.00**	0.26	9.16	0.24	7.55	0.13	4.81
Sol. AA	1.21	23.26	0.98	19.43	0.61	11.52	**0.25**	**3.68**	0.60	15.48	0.38	12.69	0.31	8.77
Water	0.76	15.27	0.83	13.57	0.72	10.31	**0.15**	**2.50**	0.59	11.69	0.47	9.83	0.32	6.89
QMugs	0.69	23.58	0.54	16.93	0.41	8.74	**0.12**	**3.78**	0.34	16.17	0.32	12.90	0.17	8.66
LWAMAE[b]	0.73	15.42	0.65	11.66	0.46	5.87	**0.14**	**2.58**	0.43	10.69	0.36	8.74	0.22	5.78
# Params	2.3 M		6.9 M		45 M		6.5 M		1.3 M		2.5 M		4.9 M	
Efficiency[c] ↑	0.56		0.27		-		-		1.01		0.47		0.25	

[a] Model uses direct-force prediction.

[b] The Logarithmic Weighted Average MAE defined in Eq. (1).

[c] All available models are tested on one diamond structure with 80 atoms using one 40GB Nvidia A-800 GPU through ASE calculator.

Table S-2. Torsional MAE, RMSE and MAEB errors (all in **kcal/mol**) between MLIP predictions and its reference DFT labels on the TorsionNet-500 dataset, following Ref.[47].

Model	**Training data**	**MAE** ↓	**RMSE** ↓	**MAEB[a]**↓	**NABH$_h$[b]**↓
AIMNet2	Ref.[37]	0.38	0.55	0.58	82
DPA2-Drug	Ref.[47]	0.24	0.35	0.36	18
MACE(M)	SPICE-MACE-OFF	0.19	0.28	0.31	7
MACE(L)	SPICE-MACE-OFF	0.14	0.21	0.23	4
DPA3-L6	SPICE-MACE-OFF	0.13	0.19	0.20	3
DPA3-L12	SPICE-MACE-OFF	0.10	0.16	0.17	**0**
DPA3-L24	SPICE-MACE-OFF	**0.06**	**0.09**	**0.09**	**0**
Nutmeg(L)	SPICE2	-	-	0.20[38]	-

[a] The MAE of the torsional barrier height, defined as the difference between the minimum and the maximum energy points during the torsional rotation.

[b] The number of molecules (total:$N_{\text{mols}} = 500$) for which the model prediction of potential barrier height has an error of more than 1 kcal/mol.

Table S-3. Test RMSE on the liquid water and three ice systems[8]. Energy (E) RMSE is in **meV/H_2O**, force (F) RMSE is in **meV/Å**. Except for DeePMD, models are trained on 0.1% of the training data following Ref.[10].

	100% data		**0.1% data**									
	DeePMD		DeepPot-SE		NequIP		DPA3-L6		DPA3-L12		DPA3-L24	
System	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**
Liquid Water	1.0	40.4	1.8	66.8	1.6	51.4	**0.7**	**37.3**	1.0	41.9	1.0	44.5
Ice Ih (b)	**0.7**	43.3	4.7	61.8	2.5	57.8	1.2	32.3	1.4	**31.4**	1.3	32.3
Ice Ih (c)	**0.7**	26.8	4.9	33.4	3.9	29.1	2.6	25.2	0.9	**19.8**	1.0	20.0
Ice Ih (d)	0.8	25.4	3.2	29.2	2.6	24.1	3.2	43.4	**0.7**	**15.5**	0.8	17.9
LWARMSE[a]	**0.8**	33.0	3.4	44.8	2.5	38.0	1.6	33.9	1.0	**25.2**	1.0	26.8

[a] The Logarithmic Weighted Average RMSE defined in Eq. (1), with equal weights per subset.

Table S-4. Test MAE on three datasets from Ref.[48]. Energy (E) MAE is in **meV/atom** (except in **meV** for Zeolite), force (F) MAE is in **meV/Å**.

	NequIP		AlphaNet		DPA3-L6		DPA3-L12		DPA3-L24	
Dataset	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**
Formate D. on Cu	0.50	47.3	0.23	45.5	0.21	44.5	0.19	39.7	**0.17**	**36.9**
Defected Graphene	1.90	60.2	1.70	32.0	0.72	35.2	0.62	30.0	**0.60**	**28.2**
Zeolite[a]	-	-	**15.4**	**14.6**	29.6	23.9	22.2	19.1	16.2	15.1

[a] MAE averaged via LWAMAE defined in Eq. (1), with equal weights per subset.

Table S-5. Test RMSE on the DPA2 test sets which consist of 18 distinct datasets from Ref.[24]. Energy (E) RMSE is in **meV/atom** and Force (F) RMSE is in **meV/Å** across different models.

	Weight	GNO		eqV2		NequIP		Allegro		MACE		DPA2		DPA3-L24	
Dataset		**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**
Alloy	2.0	14.3	85.1	8.5	**62.7**	44.0	175.6	21.4	119.4	16.2	190.2	16.8	125.7	**7.1**	99.2
Cathode-P	1.0	1.5	17.9	1.1	**14.9**	14.3	69.8	1.0	24.2	2.6	37.8	0.9	24.5	**0.6**	18.6
Cluster-P	1.0	47.7	**69.6**	34.6	104.4	75.1	216.6	54.8	174.1	41.3	189.7	31.5	126.0	**29.3**	118.3
Drug	2.0	40.5	93.6	29.8	807.4	21.6	187.2	13.1	100.8	/ [b]	/	12.7	125.5	**5.5**	**54.2**
FerroEle-P	1.0	1.5	17.9	1.1	**13.0**	1.1	23.0	0.7	28.6	2.3	31.7	0.6	28.7	**0.3**	13.4
OC2M	2.0	25.0	129.1	**6.7**	**45.2**	97.4	226.1	61.3	166.8	/	/	36.2	154.0	9.0	128.0
SSE-PBE-P	1.0	2.7	**8.2**	OOM	OOM	1.6	41.1	1.0	47.8	1.8	29.9	1.4	50.3	**0.5**	19.8
SemiCond	1.0	8.0	94.4	3.9	**40.8**	20.5	180.7	6.8	146.8	12.7	182.8	5.5	123.6	**3.4**	108.3
H2O-PD	1.0	OOM[a]	OOM	OOM	OOM	0.9	27.1	OOM	OOM	79.9	29.7	0.5	24.7	**0.4**	**13.7**
Ag∪Au-PBE	0.2	106.0	8.0	23.4	**4.4**	42.3	43.8	39.2	58.9	369.1	34.5	2.4	17.8	**1.1**	10.9
Al∪Mg∪Cu	0.3	5.9	9.4	**1.9**	**5.7**	38.0	48.3	18.3	40.6	7.7	42.9	2.1	19.1	2.0	13.0
Cu	0.1	6.1	5.8	1.7	**3.8**	6.2	16.7	1.3	8.9	38.8	13.6	**1.2**	8.9	1.7	7.2
Sn	0.1	8.4	33.7	5.2	**19.6**	18.2	62.2	5.6	40.2	/	/	4.1	54.4	**2.9**	49.8
Ti	0.1	44.5	87.9	19.1	**48.6**	27.6	137.4	6.9	85.6	8.3	94.2	5.0	113.1	**4.1**	91.7
V	0.1	17.9	79.3	5.6	**47.4**	8.8	91.6	4.2	82.1	14.2	140.4	4.1	90.8	**2.8**	71.8
W	0.1	79.1	81.2	46.8	**51.3**	20.8	160.4	4.0	101.6	15.6	181.2	5.6	108.1	**2.5**	83.3
C12H26	0.1	135.8	**518.7**	123.1	907.4	121.4	715.6	140.4	648.1	81.9	802.3	55.3	692.5	**42.4**	541.6
HfO2	0.1	1.2	16.1	**1.0**	**9.1**	1.5	58.8	1.4	64.0	2.3	14.7	**1.0**	54.2	1.4	28.9
LWARMSE[c]		12.1	52.0	7.4	62.0	9.3	87.2	16.5	107.2	11.3	75.3	5.9	76.6	**2.9**	**49.8**

[a] OOM indicates Out-Of-Memory errors.

[b] / signifies an unresolved error that occurred during the training process.

[c] The Logarithmic Weighted Average RMSE defined in Eq. (1) across all systems, with weights defined in Ref.[24].

Table S-6. Hyper-parameters for DPA3 models trained on different benchmark datasets.

Hyper-parameters	SPICE -round1	SPICE -round2	MPtrj -round1	MPtrj -round2	Scaling-law	Graph-order	Others
Number of update layers L	6/12/24	6/12/24	24	24	3/6/9/16/24	6/12/24	6/12/24
Maximum graph order k	2	2	2	2	2	3	2
Dimension of vertex features in $G^{(1)}$	128	128	128	128	128	128	128
Dimension of edge features in $G^{(1)}$	64	64	64	64	64	64	64
Dimension of edge features in $G^{(2)}$	32	32	32	32	32	32	32
Dimension of edge features in $G^{(3)}$	-	-	-	-	-	32	-
Cutoff radius in $G^{(1)}$ (Å)	6.0	6.0	6.0	6.0	6.0	6.0	6.0
Cutoff radius in $G^{(2)}$ (Å)	4.0	4.0	4.0	4.0	4.0	4.0	4.0
Cutoff radius in $G^{(3)}$ (Å)	-	-	-	-	-	2.8	-
Batch size (per GPU)	$\lceil \frac{256}{N} \rceil$[a]	$\lceil \frac{256}{N} \rceil$	$\lceil \frac{128}{N} \rceil$	$\lceil \frac{128}{N} \rceil$	$\lceil \frac{128}{N} \rceil$	1	1 or $\lceil \frac{256}{N} \rceil$[d]
Number of GPUs	8	8	16	16	16	1	1
Optimizer	Adam	Adam	AdamW	AdamW	Adam	Adam	Adam
Learning rate scheduling	Exp	Exp	Exp	Exp	Exp	Exp	Exp
Maximum learning rate	1e-3	1e-3	1e-3	5e-4	1e-3	1e-3	1e-3
Minimum learning rate	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5
Number of training steps	4M	4M	2M[c]	2M[c]	2M	1M	1M
Activation Function Threshold	10.0	5.0	10.0	5.0	10.0	10.0	10.0
Loss function	MSE	Huber	MSE	Huber	MSE	MSE	MSE
Energy loss prefactor	0.2→[b]20	15	0.2→20	15	0.2→20	0.2→20	0.2→20
Force loss prefactor	100→60	1	100→20	1	100→20	100→20	100→20
Virial loss prefactor	-	-	0.02→1	2.5	0.02→1	0.02→1	0.02→1

[a] N denotes the number of atoms in each system, $\lceil \cdot \rceil$ denotes the ceiling function, which rounds the number up to the nearest integer.

[b] $\rightarrow$ indicates that the prefactors are changed in synchronization with the learning rate.

[c] We use early stopping for MPtrj training based on the validation energy MAE on the WBM test set.

[d] We use batchsize=1 for the DPA2 test sets for fair comparison and $\lceil \frac{256}{N} \rceil$ for other benchmark tests.

References

1. Schrödinger, E. Quantisierung als eigenwertproblem. *Annalen der physik* **386**, 109–139 (1926).
2. Born, M. & Heisenberg, W. Zur quantentheorie der molekeln. *Orig. Sci. Pap. Wissenschaftliche Orig.* 216–246 (1985).
3. Frenkel, D. & Smit, B. *Understanding Molecular Simulation: From Algorithms to Applications*, vol. 1 (Elsevier, 2001).
4. Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Phys. review* **136**, B864 (1964).
5. Kohn, W. & Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. review* **140**, A1133 (1965).
6. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. review letters* **98**, 146401 (2007).
7. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. review letters* **104**, 136403 (2010).
8. Zhang, L., Han, J., Wang, H., Car, R. & E, W. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. review letters* **120**, 143001 (2018).
9. Schütt, K. *et al.* Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Adv. neural information processing systems* **30** (2017).
10. Batzner, S. *et al.* E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. communications* **13**, 2453 (2022).
11. Gasteiger, J., Becker, F. & Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Adv. Neural Inf. Process. Syst.* **34**, 6790–6802 (2021).
12. Batatia, I., Kovacs, D. P., Simm, G., Ortner, C. & Csányi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Adv. Neural Inf. Process. Syst.* **35**, 11423–11436 (2022).
13. Jia, W. *et al.* Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. In *SC20: International conference for high performance computing, networking, storage and analysis*, 1–14 (IEEE, 2020).
14. Chen, C. & Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nat. Comput. Sci.* **2**, 718–728 (2022).
15. Zhou, G. *et al.* Uni-mol: A universal 3d molecular representation learning framework. *ChemRxiv* (2022).
16. Choudhary, K. *et al.* Unified graph neural network force-field for the periodic table: solid state applications. *Digit. Discov.* **2**, 346–355 (2023).
17. Deng, B. *et al.* Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nat. Mach. Intell.* **5**, 1031–1041 (2023).
18. Merchant, A. *et al.* Scaling deep learning for materials discovery. *Nature* 1–6 (2023).
19. Batatia, I. *et al.* A foundation model for atomistic materials chemistry. *arXiv preprint arXiv:2401.00096* (2023).

20. Yang, H. *et al.* Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967* (2024).

21. Rhodes, B. *et al.* Orb-v3: atomistic simulation at scale. *arXiv:2504.06231* (2025).

22. Mazitov, A. & et.al. Pet-mad, a universal interatomic potential for advanced materials modeling. *arXiv:2503.14118* (2025).

23. Shoghi, N. *et al.* From molecules to materials: Pre-training large generalizable models for atomic property prediction. In *The Twelfth International Conference on Learning Representations* (2024).

24. Zhang, D. *et al.* Dpa-2: a large atomic model as a multi-task learner. *npj Comput. Mater.* **10**, 293 (2024).

25. Wang, R., Gao, Y., Wu, H. & Zhong, Z. Pfd: Automatically generating machine learning force fields from universal models. *arXiv preprint arXiv:2502.20809* (2025).

26. Peng, A. *et al.* Lambench: A benchmark for large atomic models. *arXiv preprint arXiv:2504.19578* (2025).

27. Kaplan, J. *et al.* Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).

28. Bi, X. *et al.* Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954* (2024).

29. Su, H., Tian, Z., Shen, X. & Cai, X. Unraveling the mystery of scaling laws: Part i. *arXiv preprint arXiv:2403.06563* (2024).

30. Yuan, E. C.-Y. *et al.* Foundation models for atomistic simulation of chemistry and materials. *arXiv preprint arXiv:2503.10538* (2025).

31. Ji, X. *et al.* Uni-mol2: Exploring molecular pretraining model at scale. *arXiv preprint arXiv:2406.14969* (2024).

32. Li, C. *et al.* Scaling laws of graph neural networks for atomistic materials modeling. *arXiv preprint arXiv:2504.08112* (2025).

33. Barroso-Luque, L. *et al.* Open materials 2024 (omat24) inorganic materials dataset and models. *arXiv preprint arXiv:2410.12771* (2024).

34. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. review letters* **77**, 3865 (1996).

35. Jain, A. *et al.* The materials project: A materials genome approach to accelerating materials innovation. *APL Mater* (2013).

36. Kovács, D. P. *et al.* Mace-off23: Transferable machine learning force fields for organic molecules. *arXiv preprint arXiv:2312.15211* (2023).

37. Anstine, D., Zubatyuk, R. & Isayev, O. Aimnet2: a neural network potential to meet your neutral, charged, organic, and elemental-organic needs. *arXiv preprint* (2024).

38. Eastman, P., Pritchard, B. P., Chodera, J. D. & Markland, T. E. Nutmeg and spice: models and data for biomolecular machine learning. *J. chemical theory computation* **20**, 8583–8593 (2024).

39. Kim, J. *et al.* Data-efficient multifidelity training for high-fidelity machine learning interatomic potentials. *J. Am. Chem. Soc.* **147**, 1042–1054, DOI: 10.1021/jacs.4c14455 (2024).

40. Tuckerman, M. *Statistical Mechanics: Theory and Molecular Simulation* (OUP Oxford, 2010).

41. Fu, X. *et al.* Learning smooth and expressive interatomic potentials for physical property prediction. *arXiv preprint arXiv:2502.12147* (2025).

42. The openlam-v1 dataset (2025). https://www.aissquare.com/datasets/detail?name=OpenLAM-TrainingSet-v1&id=308&pageType=datasets.

43. Whitney, H. Congruent graphs and the connectivity of graphs. *Hassler Whitney Collect. Pap.* 61–79 (1992).

44. Krausz, J. Démonstration nouvelle d'une théoreme de whitney sur les réseaux. *Mat. Fiz. Lapok* **50**, 75–85 (1943).

45. Harary, F. & Norman, R. Z. Some properties of line digraphs. *Rendiconti del circolo matematico di palermo* **9**, 161–168 (1960).

46. Rai, B. K. *et al.* Torsionnet: A deep neural network to rapidly predict small-molecule torsional energy profiles with the accuracy of quantum mechanics. *J. Chem. Inf. Model.* **62**, 785–800 (2022).

47. Yang, M. *et al.* Ab initio accuracy neural network potential for drug-like molecules. *chemrxiv preprint* (2024).

48. Yin, B. *et al.* Alphanet: Scaling up local frame-based atomistic foundation model. *arXiv preprint arXiv:2501.07155* (2025).

49. Najibi, A. & Goerigk, L. The nonlocal kernel in van der waals density functionals as an additive correction: An extensive analysis with special emphasis on the b97m-v and ωb97m-v approaches. *J. Chem. Theory Comput.* **14**, 5725–5738 (2018).

50. Weigend, F. & Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for h to rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **7**, 3297–3305 (2005).

51. Rappoport, D. & Furche, F. Property-optimized gaussian basis sets for molecular response calculations. *The J. chemical physics* **133** (2010).

52. Grimme, S., Ehrlich, S. & Goerigk, L. Effect of the damping function in dispersion corrected density functional theory. *J. computational chemistry* **32**, 1456–1465 (2011).

53. Grimme, S., Antony, J., Ehrlich, S. & Krieg, H. A consistent and accurate ab initio parametrization of density functional dispersion correction (dft-d) for the 94 elements h-pu. *The J. chemical physics* **132** (2010).

54. Qu, E. & Krishnapriyan, A. S. The importance of being scalable: Improving the speed and accuracy of neural network interatomic potentials across chemical domains. *arXiv preprint arXiv:2410.24169* (2024).

55. 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** (2018).

56. Riebesell, J. *et al.* Matbench discovery–a framework to evaluate machine learning crystal stability predictions. *arXiv preprint arXiv:2308.14920* (2023).

57. Wang, H., Botti, S. & Marques, M. Predicting stable crystalline compounds using chemical similarity. *npj Comput. Mater.* **7**, 12, DOI: 10.1038/s41524-020-00481-6 (2021).

58. Chanussot, L. *et al.* Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catal.* **11**, 6059–6072 (2021).

59. Sours, T. G. & Kulkarni, A. R. Predicting structural properties of pure silica zeolites using deep neural network potentials. *The J. Phys. Chem. C* **127**, 1455–1463 (2023).

60. Neumann, M. *et al.* Orb: A fast, scalable neural network potential (2024). 2410.22570.

61. Rhodes, B. *et al.* Orb-v3: atomistic simulation at scale (2025). 2504.06231.

62. Kim, J. *et al.* Data-efficient multifidelity training for high-fidelity machine learning interatomic potentials. *J. Am. Chem. Soc.* **147**, 1042–1054 (2024).

63. Bochkarev, A., Lysogorskiy, Y. & Drautz, R. Graph atomic cluster expansion for semilocal interactions beyond equivariant message passing. *Phys. Rev. X* **14**, 021036 (2024).

64. Levine, D. S. *et al.* The open molecules 2025 (omol25) dataset, evaluations, and models. *arXiv preprint arXiv:2505.08762* (2025).

65. Christensen, A. S. & Von Lilienfeld, O. A. On the role of gradients for machine learning of molecular energies and forces. *Mach. Learn. Sci. Technol.* **1**, 045018 (2020).

66. Kovács, D. P. *et al.* Linear atomic cluster expansion force fields for organic molecules: beyond rmse. *J. chemical theory computation* **17**, 7696–7711 (2021).

67. Batatia, I. *et al.* The design space of e (3)-equivariant atom-centered interatomic potentials. *arXiv preprint arXiv:2205.06643* (2022).

68. Eastman, P. *et al.* Spice, a dataset of drug-like molecules and peptides for training machine learning potentials. *Sci. Data* **10**, 11 (2023).

69. Bannwarth, C., Ehlert, S. & Grimme, S. Gfn2-xtb—an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *J. chemical theory computation* **15**, 1652–1671 (2019).

70. Isert, C., Atz, K., Jiménez-Luna, J. & Schneider, G. Qmugs, quantum mechanical properties of drug-like molecules. *Sci. Data* **9**, 273 (2022).

71. Smith, D. G. *et al.* Psi4 1.4: Open-source software for high-throughput quantum chemistry. *The J. chemical physics* **152** (2020).

72. Smith, J. S., Nebgen, B., Lubbers, N., Isayev, O. & Roitberg, A. E. Less is more: Sampling chemical space with active learning. *The J. chemical physics* **148**, 241733 (2018).

73. Kühne, T. D. *et al.* Cp2k: An electronic structure and molecular dynamics software package-quickstep: Efficient and accurate electronic structure calculations. *The J. Chem. Phys.* **152** (2020).

74. Ying, P., Natan, A., Hod, O. & Urbakh, M. Effect of interlayer bonding on superlubric sliding of graphene contacts: A machine-learning potential study. *ACS nano* **18**, 10133–10141 (2024).

75. Kresse, G. & Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. materials science* **6**, 15–50 (1996).

76. Kresse, G. & Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. review B* **54**, 11169 (1996).

77. Deep modeling community (2017). https://deepmodeling.com.

78. Frisch, M. *et al.* Gaussian 16 (2016).

79. Chen, M., Guo, G. & He, L. Systematically improvable optimized atomic basis sets for ab initio calculations. *J. Physics: Condens. Matter* **22**, 445501 (2010).

80. Li, P. *et al.* Large-scale ab initio simulations based on systematically improvable atomic basis. *Comput. Mater. Sci.* **112**, 503–517 (2016).

81. Jain, A. *et al.* Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials* **1**, 011002 (2013).