Supplementary information

Technical details of PFD-kit

The PFD workflow is implemented using the dflow package [1], a Python-based cloud native workflow package for scientific applications. Within the dflow framework, each step in the PFD can be run in an isolated container node sequenced by the workflow server. The dflow framework enables efficient parallel execution of time-consuming steps such as first-principles calculations and MD explorations. It also allows for the decoupling between the computation resources and the settings of the software environment. To save development effort, some components of the PFD workflow, such as LAMMPS and VASP modules, are reused from DPGEN2, an active learning scheme to generate DeePMD potentials also based on the dflow package [2, 3].

The PFD-kit has two major modules, the fine-tuning and distillation workflows. In both workflows, initial structures need to be provided, and it is randomly perturbed to construct a small set of configurations that serve as the starting point of exploration or as an initial training dataset. Possible perturbation operations include lattice contraction and atomic displacement, whose parameters can be specified by the users. Figure S1 shows the design of the iterative fine-tuning workflow. In each iteration, the fine-tuned model from the last iteration would run short MD simulations (usually in the range of ps) to generate new structures. The newly generated structures are than being filtered based on a set of criterion such as interatomic distances, and subsequently being labeled using DFT first-principles calculations. The labeled dataset is then tested for the fine-tuned model. If converged, the fine-tuned model would be output, and the fine-tuing process ends; if not, some of the labeled dataset would be added to the training dataset, and train a new model by fine-tuning a base model, which could either be the fine-tuned model from the last iteration or the pre-trained foundation model.

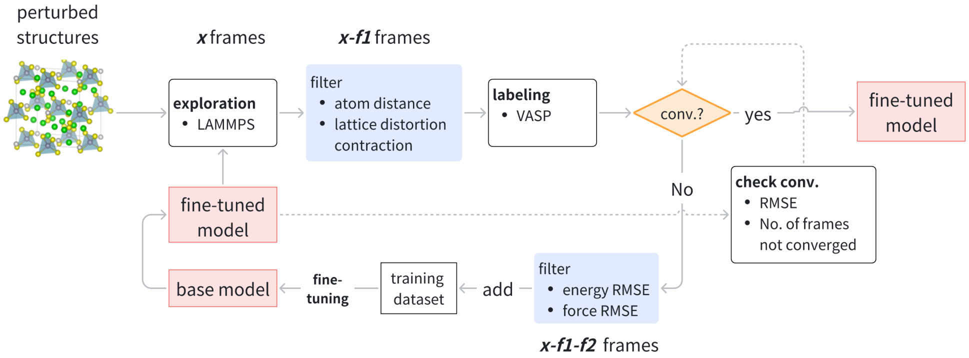

Figure S1: Detailed workflow of the fine-tuning module.

Contrary to fine-tuning, in distillation workflow, the training dataset can be generated in a high-throughout manner because of the much lower cost of the fine-tuned model compared with DFT calculations. With a very large energy and force dataset labeled using the fine-tuned model, a simpler model, in this case a standard DeePMD model with local descriptor, can be trained, and there is generally no need for iterative training. The code repository and the documentation of PFD-kit can be found at https://github.com/ruoyuwang1995nya/pfd-kit.

Hardware specifications

All the tests are carried out using the Bohrium online platform. The type of GPU devices used for model training and MD simulation with DPA-2 and DeePMD model is Nvidia V100 card of 32 GB memory; the CPU nodes used for DFT calculations have 32 CPU cores. The time consumption of the PFD workflow is estimated using the above hardware specifications and assuming parallel execution of DFT and MD simulations.

Data generation

All DFT datasets are generated using the VASP package with the PBE pseudopotential and plane-wave basis functions except for the $\mathrm{Li}_{1+x}\mathrm{Al}_x\mathrm{Ti}_{2-x}(\mathrm{PO}_4)_3$ solid electrolytes, which is calculated using the open-source ABACUS software with linear combination of atomic orbital (LCAO) basis functions.

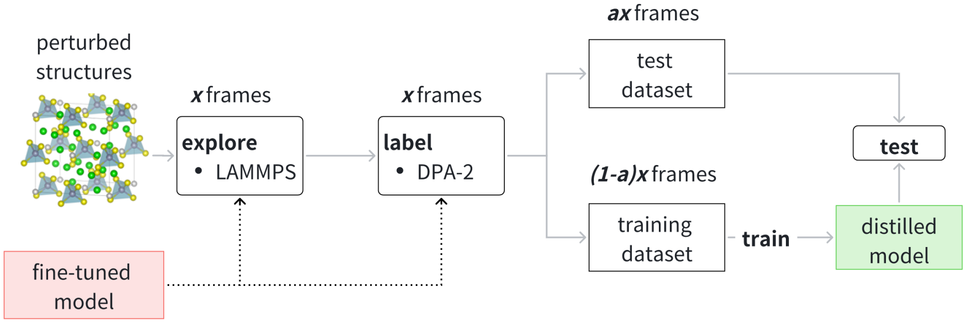

Figure S2: Detailed workflow of the distillation module.

The training dataset for crystalline silicon consists of 2x2x2 supercells each having 64 atoms. The configurations in the $Li_{1+x}Al_xTi_{2-x}(PO_4)_3$ (LATP) training set each have 110 atoms. LATP structures of various Li compositions and Al substitution are built using the Pymatgen package. For the example of 1,4-polyisoprene molecular chains, each chain is made up of carbon and oxygen atoms with a total number of atoms ranging from 40 ($C_{20}H_{20}$) up to 180 ($C_{60}H_{120}$). The test set is constructed by clusters of several molecular chains. The amorphous carbon structures with sp^3 hybridization are randomly generated using the RG^2 package [4], and the number of carbon atoms in the unit cell ranges from 8 up to 84. The training set for high entropy perovskite $Ba_{0.5}Na_{0.25}Bi_{0.25}Ti_{0.875}Zr_{0.125}O_3$ is constructed by random substitution in a 2x2x2 basic $BaTiO_3$ supercell of 40 atoms. The initial structures of the Li_6PSCl_5/Li interfaces are made by first concatenating the Li crystal and the bulk Li_6PSCl_5 structure and then running structural optimization. Each interface structure contains 106 atoms.

Additional results

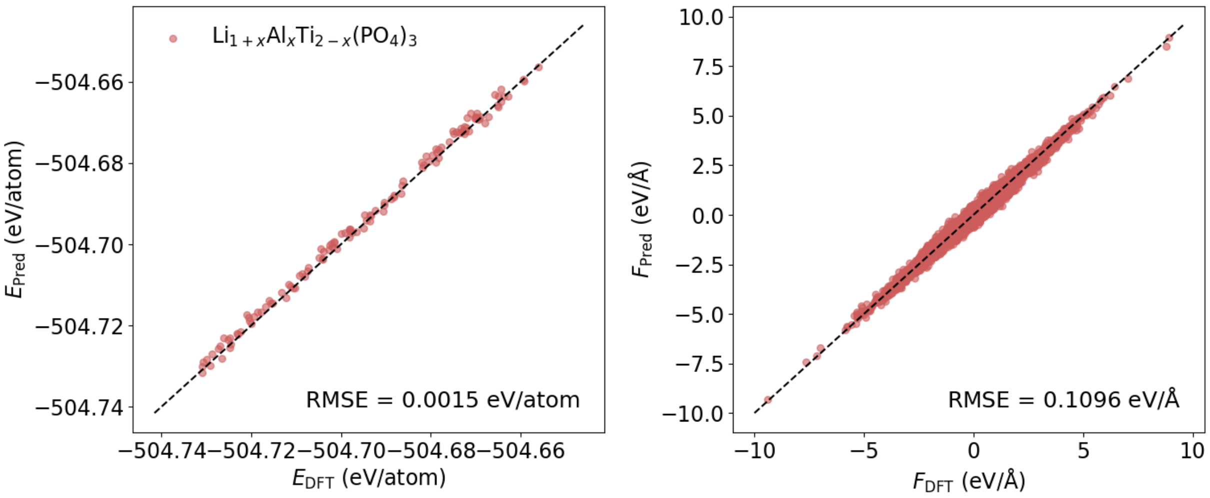

Figure S3: Energy and force prediction the distill model for $Li_{1+x}Al_xTi_{2-x}(PO_4)_3$ solid electrolyte.

Table S1: Training cost and accuracy of the fine-tuned and distilled models generated using the PFD workflow for the $Li_{1+x}Al_xTi_{2-x}(PO_4)_3$ solid electrolyte.

	Fine-tuning	Distillation
Data size	138	1667
Number of iterations	3	1
Energy RMSE (eV/atom)	0.0013	0.0015
Force RMSE (eV/Å)	0.051	0.1096

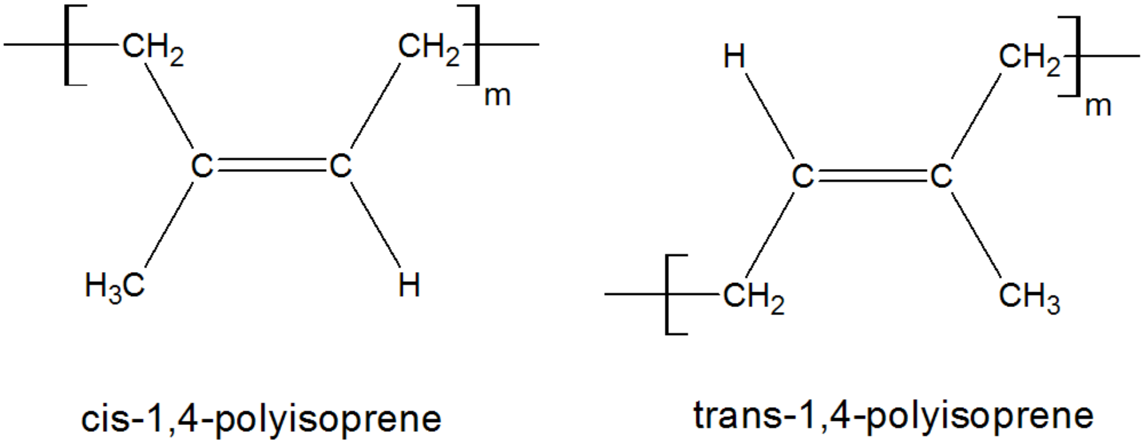

Figure S4: The chemical formula of cis- and trans-1,4-polyisoprene chains.

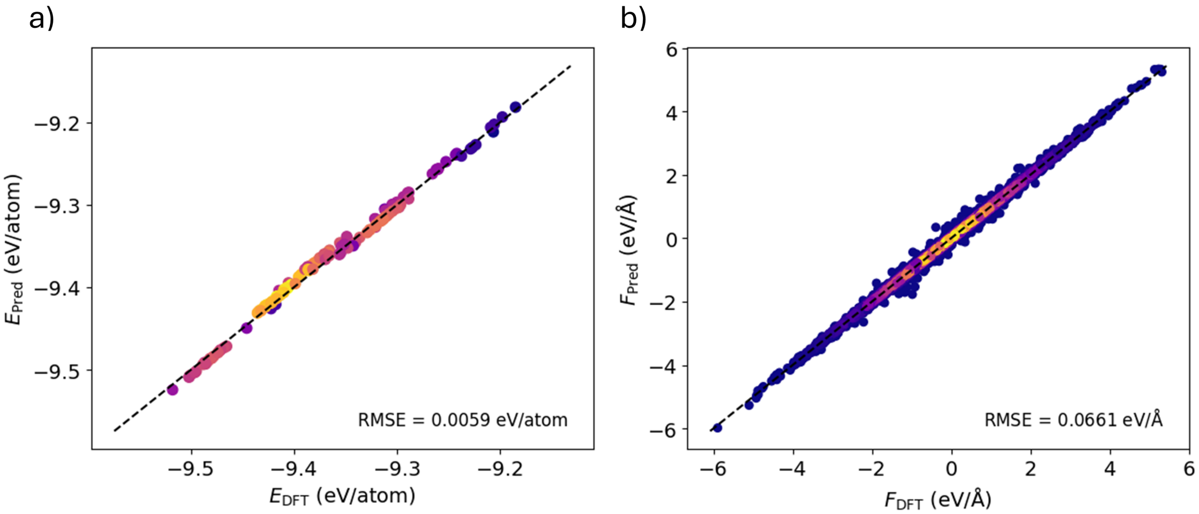

Figure S5: **Fine-tuned model for amorphous carbon generated using PFD workflow.** The **a)** energy and **b)** force prediction error of the fine-tuned model on amorphous carbon of sp^3 tetrahedral local geometry.

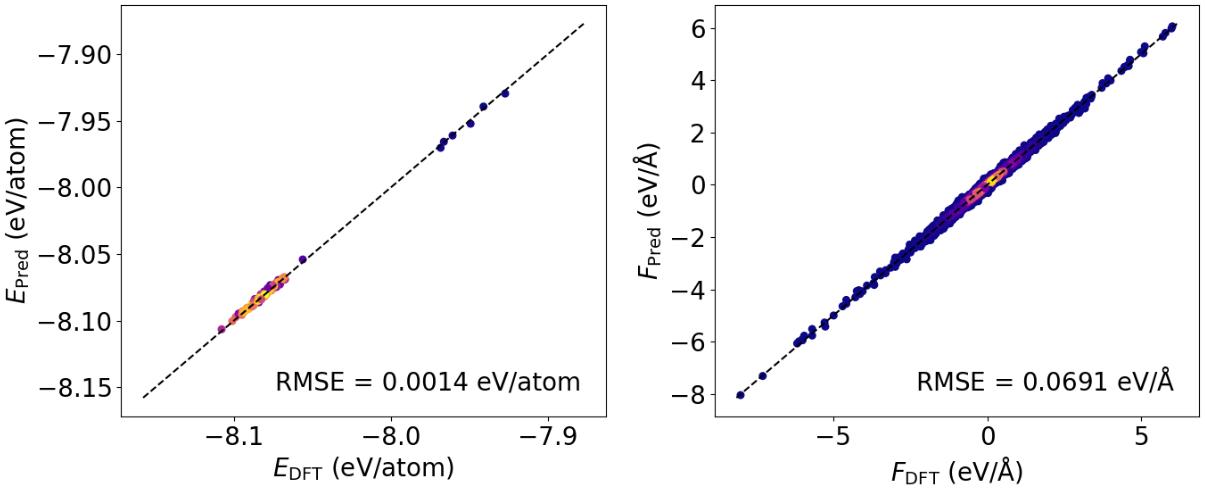

Figure S6: Energy and force prediction accuracy of the distilled model for the doped perovskite $\mathrm{Ba_{0.5}Na_{0.25}Bi_{0.25}Ti_{0.875}Zr_{0.125}O_3}$.

Table S2: Doped $\mathrm{Ba_{0.5}Na_{0.25}Bi_{0.25}Ti_{0.875}Zr_{0.125}O_3}$ model generated using PFD workflow

	Fine-tuning	Distillation
Data size	134[a]	3818
Number of iterations	2	1
Time	5 h	3 h
Energy RMSE (eV/atom)	0.0010	0.0014
Force RMSE (eV/Å)	0.0472	0.0691

[a] In total, 242 DFT calculations are performed, including test set.

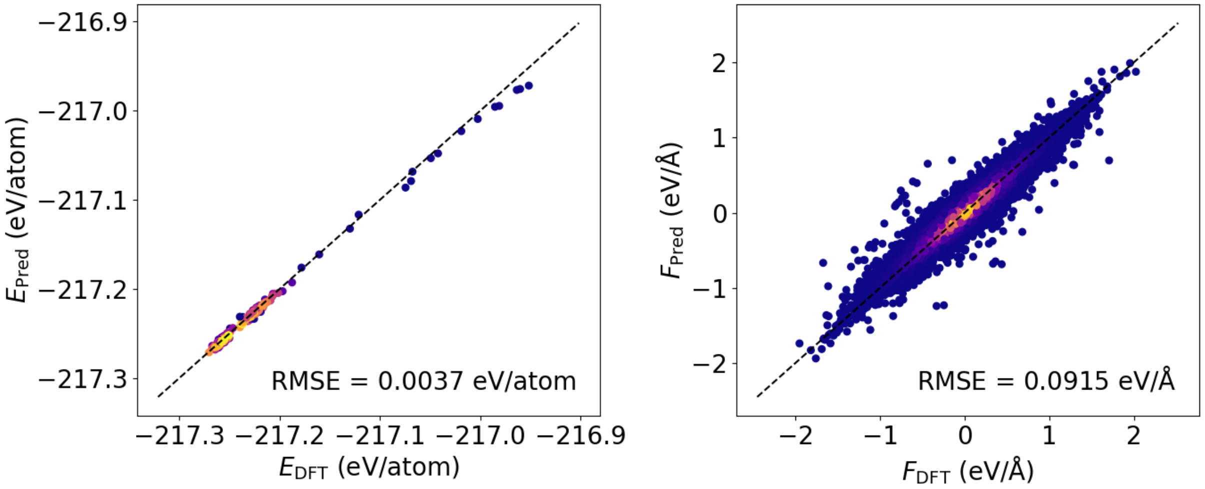

Figure S7: Energy and force accuracy of the distilled model for the $\mathrm{Li_6PSCl_5/Li}$ interface model.

References

[1] Xinzijian Liu, Yanbo Han, Zhuoyuan Li, Jiahao Fan, Chengqian Zhang, Jinzhe Zeng, Yifan Shan, Yannan Yuan, Wei-Hong Xu, Yun-Pei Liu, Yuzhi Zhang, Tongqi Wen, Darrin M. York, Zhicheng Zhong, Hang Zheng, Jun Cheng, Linfeng Zhang, and Han Wang. Dflow, a Python framework for constructing cloud-native AI-for-Science workflows, April 2024. arXiv:2404.18392 [cs].

[2] Linfeng Zhang, De-Ye Lin, Han Wang, Roberto Car, and Weinan E. Active learning of uniformly accurate interatomic potentials for materials simulation. *Physical Review Materials*, 3(2):023804, February 2019.

[3] Yuzhi Zhang, Haidi Wang, Weijie Chen, Jinzhe Zeng, Linfeng Zhang, Han Wang, and Weinan E. DP-GEN: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, 253:107206, August 2020.

[4] Xizhi Shi, Chaoyu He, Chris J. Pickard, Chao Tang, and Jianxin Zhong. Stochastic generation of complex crystal structures combining group and graph theory with application to carbon. *Physical Review B*, 97(1):014104, January 2018.