Efficient Universal Potential Distillation with Pre-trained Students in LightPFP

Wenwen Li
Preferred Networks
wenwenli@preferred.jp

Nontawat Charoenphakdee
Preferred Networks
nontawat@preferred.jp

Yong-Bin Zhuang
Preferred Networks
ybzhuang@preferred.jp

Yuta Tsuboi
Preferred Networks
tsuboi@preferred.jp

Ryuhei Okuno
Preferred Networks
ok79ryuhei@preferred.jp

So Takamoto
Preferred Networks
takamoto@preferred.jp

Abstract

Machine learning interatomic potentials (MLIPs) bridge the gap between the accuracy of quantum mechanics and the efficiency of classical simulations. Although universal MLIPs (u-MLIPs) offer broad transferability across diverse chemical spaces, their high inference costs limit their scalability in large-scale simulations. In this paper, we propose *LightPFP*, a knowledge distillation framework designed to train computationally efficient task-specific MLIPs (ts-MLIPs) tailored for specific systems by leveraging of u-MLIPs. Unlike prior approaches that only pre-trains u-MLIPs on large datasets, LightPFP incorporates an additional step where student models are pre-trained as well. This dual pre-training strategy significantly enhances the data efficiency of the student models, enabling them to achieve higher performance with limited training data. We validate the effectiveness of LightPFP using Ni_3Al Alloy simulation, showcasing its data efficiency, and further compare its performance against other methods in estimating the mechanical and grain boundary properties of AlCoCrFeNi high-entropy alloy.

1 Introduction

The development of accurate and efficient machine learning potentials (MLIPs), typically trained on data labeled by density functional theory (DFT), is critical for enabling large-scale atomistic simulations in materials science and chemistry. MLIPs can be broadly classified into two paradigms: universal MLIPs (u-MLIPs) and task-specific MLIPs (ts-MLIPs). The former, e.g., M3GNet [1], CHGNet [2], Matlantis PFP [3, 4], MACE [5, 6], are trained on vast datasets spanning millions of diverse structures, covering broad regions of chemical and configurational space. They provide remarkable capabilities out-of-the-box transferability. In particular, Matlantis PFP, trained on a diverse and complex DFT database, has been shown to demonstrate its robustness by achieving high performance across a wide range of materials without fine-tuning, including battery [7–12], MOF [13, 14], ceramics [15, 16], catalyst[17], polymer [18], nanotube [19], atomic layer deposition [20, 21], Hydrogen storage [22], superconductor [23], Memristor [24]. Despite their versatility, the large model sizes of u-MLIPs, required to capture diverse chemical systems, can limit their computational efficiency in large-scale molecular dynamics (MD) simulations. In contrast, ts-MLIPs use simpler architectures that trade universality for greater computational efficiency. However, the development of ts-MLIPs remains resource-intensive, requiring significant time and effort for DFT-based dataset generation and model training.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Accelerated Materials Discovery (AI4Mat).

Perspective	[25]	[26]	[27]	[28]	LightPFP (This work)
Teacher is trained from multiple tasks	×	✓	✓	✓	✓
Use teacher in data generation	✓	×	✓	✓	✓
Enable active learning with teacher's labels	×	×	✓	✓	✓
Use student pre-training	×	×	×	×	✓
Does not require teacher's fine-tuning	×	×	×	×	✓

Table 1: Comparisons of the proposed method with existing works [25–28].

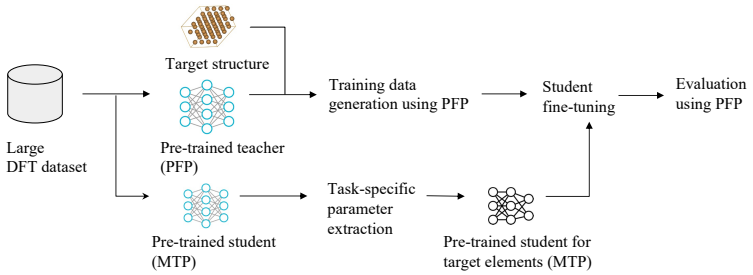

Figure 1: Schematic diagram of LightPFP.

To address the challenges associated with training ts-MLIPs, knowledge distillation has been introduced as a strategy to train more efficient MLIPs (referred to as "students") by leveraging knowledge from larger, more general MLIPs (referred to as "teachers") [27, 29, 25, 28, 30, 26]. For instance, Zhang et al. [28] proposed the DPA-2 framework, where the teacher model is pre-trained on multiple tasks, similar to the u-MLIP training paradigm. The pre-trained teacher is then finetuned on task-specific DFT-labeled data by modifying its descriptor and fitting networks. Finally, active learning is employed, iteratively using MD simulations of the finetuned teacher model to gather new data for training the student model. Similarly, Gardner et al. [27] explored distillation from foundation models (e.g., MatterSim [31], MACE [5], OrbNet [32]) into task-specific MLIPs (e.g., PaiNN [33], ACE [34]). Their approach involves first conducting fine-tuning on the teacher (u-MLIP) model, followed by using the teacher to generate task-specific data for training the student model.

In this paper, we propose the *LightPFP* framework, which utilizes the PFP [3] as a u-MLIP teacher *without fine-tuning* to generate training data. Table 1 highlights the key distinctions between this study and prior work. To the best of our knowledge, previous studies have largely relied on fine-tuning the u-MLIP as a prerequisite for distillation. Furthermore, the role of student pre-training in enhancing data efficiency has received limited attention in existing literature. In the evaluation, we demonstrate the benefits of student pre-training through simulations of Ni_3Al alloy systems. Subsequently, we compare the performance of LightPFP to MTP trained with DFT data, PFP, and MACE-MP-0, to estimate mechanical and grain boundary properties of AlCoCrFeNi high-entropy alloy.

2 LightPFP

In this section, overview of LightPFP is presented. For the teacher model, we employ PFP [3] based on TeaNet architecture [35]. PFP and LightPFP are both available in Matlantis [30]. PFP serves as the source for generating training data, which forms the foundation for training the student model in LightPFP. As the student model, we adopt the Moment Tensor Potential (MTP), proposed by Novikov et al. [36], due to its favorable trade-off between accuracy and efficiency [37]. Figure 1 shows the workflow of LightPFP, which begins by defining a target structure and preparing training data using the PFP u-MLIP framework. Following a similar approach to PFP, pre-trained students are trained on diverse chemical systems dataset described in Takamoto et al. [3]. Reptile algorithm [38] is used for student pre-training (see Appendix B for details). Due to the large dataset's size, student pre-training is expected to take some time. Nevertheless, a pre-trained student can be prepared in advance. LightPFP in Matlantis [4] offers several pre-trained students readily available. MTP contains learned parameters specific to each element pair. To reduce the model size, we select only the subset of parameters from the pre-trained MTP that corresponds to the elements present in the target structure, since other parameters are never utilized and can therefore be omitted. The extracted MTP is then fine-tuned using PFP-generated data, after which its performance is evaluated.

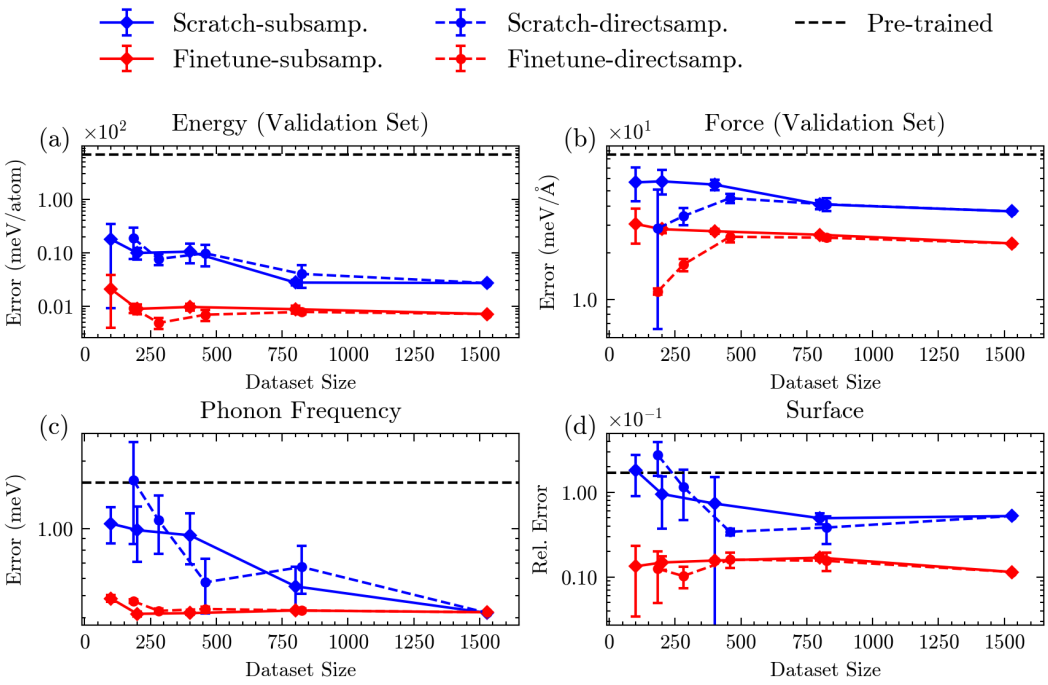

Figure 2: Comparison of data efficiency between finetuned pre-trained and scratch-trained student models.

3 Results

3.1 Ni_3Al alloy

We first demonstrate the enhanced data efficiency of pre-trained student models, using the Ni_3Al alloy [39] as an example. To this end, a full dataset containing 1529 structures is prepared through the comprehensive sampling involving PFP [3] in the relevant configuration space. The sampling methods comprise static and dynamic samplings. The static method samples static structures by compressing and deforming its lattice, as well as displacing atomic positions. The dynamic sampling uses MD simulations with initial configurations of both defect-free and defective bulk structures, as well as surface structures. The details of sampling parameters are provided in Appendix. C.1 For testing data efficiency, smaller datasets with sizes ranging from 100 to 850 are created by two methods, subsampling from the full dataset and direct sampling through the decrease of MD steps. Each size of datasets are created five times to obtain the uncertainties of errors. Structures in the datasets obtained by subsampling tend to be more widely distributed in configuration space, whereas direct sampling is closer to common user practice in real situations (i.e. by decreasing MD steps).

We compare the performance of finetuned pre-trained and scratch-trained student models on energy and force errors, as shown in Figure 2a,b. Across all dataset sizes, the finetuned pre-trained student models outperform the scratch-trained student models. We note that finetuning pre-trained student models on 100 structures performs almost as well as on 1529 structures. In addition to the standard energy and force testings, we validate the performance of student models on different application tasks, for instance, phonon spectra and surface energies, as shown in Figure 2c,d. Comparable to the energy and force testings, the performance of finetuned pre-trained models is better than the scratch-trained models. Similar performance trend can be observed in properties (see Appendix C.4).

Moreover, the performance of finetuned pre-trained student models are more robust in application tasks whereas scratch-trained models show typical overfitting behavior. The force errors from the scratch-trained models on the smaller datasets (Figure 2b) are lower than on the larger dataset. However, the errors on phonon spectra and surface energies are larger (Figure 2c,d). In contrast, although finetuned pre-trained student models show a similar trend on force testing, their performance on application tasks are consistently reliable across various dataset sizes.

Table 2: Comparison of DFT and MLIPs on HEA property calculation

Property	DFT	PFP	LightPFP	MACE-MP-0	MTP-DFT
Equation of State					
Volume (Å^3/atom)	11.58	**11.51**	*11.51*	11.48	11.29
Bulk modulus (GPa)	165.64	**165.66**	*164.35*	159.18	162.27
Mechanical Properties (GPa)					
Bulk modulus	159.23	165.45	167.81	157.14	169.02
Shear modulus	69.99	65.79	60.42	45.05	58.54
Young's modulus	183.14	174.27	161.84	123.36	157.44
Average Error	–	**6.43**	*6.72*	21.5	19.82
Surface Energy (eV/Å^2)					
Average Error	–	0.0058	0.0053	*0.0052*	**0.0036**
Grain Boundary Energy (eV/Å^2)					
Average Error	–	*0.0081*	**0.0063**	0.0207	0.0095

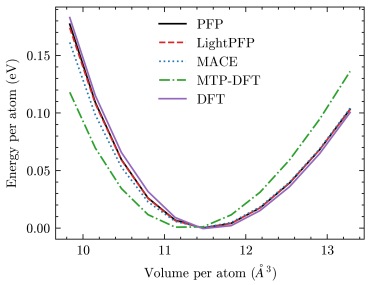

Figure 3: Equation of states of HEA calculated by different methods.

3.2 High entropy alloys (HEAs)

We use the Cantor alloy with a face-centered cubic (FCC) lattice, where the composition is 20% each of Al, Co, Cr, Fe, and Ni. HEAs have attracted attention due to their exceptional mechanical properties. We employ four models for simulation: PFP [3], MACE-MP-0 [6], MTP trained from DFT data (MTP-DFT), and LightPFP. Note that LightPFP and MTP-DFT architectures are identical, resulting in the same inference speed. Data generation for LightPFP using PFP took 24 hours, while MTP-DFT required 637 hours on a single GPU for PFP-driven molecular dynamics sampling with DFT labeling. Using ab-initio MD sampling for MTP-DFT would have taken an estimated 60,000 hours, rendering MLIP construction prohibitively expensive. Details of the data generation process are provided in Appendix D.1. Both LightPFP and MTP-DFT share a similar training time of 1 hour.

Computational efficiency: We benchmarked the molecular dynamics performance on an Nvidia V100 GPU (16GB). LightPFP and MTP-DFT achieved simulation speeds of 9.8×10^{-7} s/step/atom, which is 66 times faster than PFP (6.5×10^{-5} s/step/atom) and 249 times faster than MACE-MP-0 (2.44×10^{-4} s/step/atom). The maximum number of atoms that can be simulated on a single GPU was 716,800—52 times more than PFP (13,824) and 400 times more than MACE-MP-0 (1,792). MLIPs obtained through LightPFP significantly outperform universal potentials in computational efficiency.

Property calculation accuracy: We computed important properties of the HEA using four models, where DFT serves as a ground truth. Calculation details are provided in Appendix D.2. Results of equation of state calculation are shown in Figure 3, and equilibrium volume and bulk modulus are listed in Table 2. PFP, LightPFP and MACE-MP-0 have a good agreement with DFT in energy-volume curve, while MTP-DFT underestimate the equilibrium volume by 2.5%, possibly due to the insufficient dataset. For elastic properties, PFP, LightPFP, and MTP-DFT closely matched DFT, while MACE-MP-0 showed larger deviations. This could be due to MACE-MP-0 training data are mainly equilibrium structures and this may cause potential energy surface softening effect [40]. Next, the surface energies of seven different crystal planes were evaluated, where the crystal planes are not included in the training dataset. All models achieved high accuracy, with errors < 0.006 eV/Å^2. In terms of average error, performance ranking in descending order is as follows: MTP-DFT, MACE, LightPFP, and PFP. Finally, we computed formation energies of five grain boundaries (GB) that are not included in training dataset. LightPFP, PFP, and MTP-DFT demonstrated strong accuracy, with errors below 0.01 eV/Å^2, whereas MACE-MP-0 exhibited a larger error. Based on average error, the models are ranked in descending order of performance as follows: LightPFP, PFP, MTP-DFT, and MACE-MP-0. Across multiple properties, LightPFP achieved competitive accuracy to other methods.

4 Conclusions

We have introduced LightPFP, a distillation framework built on the PFP universal potential. By utilizing data generated by PFP and integrating a pre-trained student model, LightPFP achieved significant improvements in data efficiency and inference speed while maintaining high accuracy. Limitations of this study include the use of only the MTP model as the student and PFP as the teacher, without exploring alternative choices for either role. In addition, future work could investigate broader applications and examine different training techniques for the distillation method.

References

[1] Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.

[2] Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J Bartel, and Gerbrand Ceder. CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

[3] So Takamoto, Chikashi Shinagawa, Daisuke Motoki, Kosuke Nakago, Wenwen Li, Iori Kurata, Taku Watanabe, Yoshihiro Yayama, Hiroki Iriguchi, Yusuke Asano, et al. Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements. *Nature Communications*, 13(1):2991, 2022.

[4] Matlantis, software as a service style material discovery tool. `https://matlantis.com/`.

[5] Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35:11423–11436, 2022.

[6] Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M Elena, Dávid P Kovács, Janosh Riebesell, Xavier R Advincula, Mark Asta, Matthew Avaylon, William J Baldwin, et al. A foundation model for atomistic materials chemistry. *arXiv preprint arXiv:2401.00096*, 2023.

[7] Songjia Kong, Naoki Matsui, Satoshi Hori, Masaaki Hirayama, Kazuhiro Mori, Takashi Saito, Ryoji Kanno, and Kota Suzuki. Exploration of lithium-ion conductors based on local coordination environments using crystallographic site fingerprints. *Journal of the American Chemical Society*, 2025.

[8] Yoyo Hinuma and Mitsunori Kitta. Facile formation of two-phase domains in a single crystalline Li7–x Ti5o12 particle. *ACS Applied Energy Materials*, 2025.

[9] Shunsuke Narumi, H Eugenio Otal, Tien Quang Nguyen, Michihisa Koyama, and Nobuyuki Zettsu. Tailoring the room-temperature miscibility gap in ordered spinel LiNi 0.5 Mn 1.5 o 4 cathodes by multi-element doping. *Journal of Materials Chemistry A*, 2025.

[10] Beom Gwon Son, Choah Kwon, YongJun Cho, Taegyu Jang, Hye Ryung Byon, Sangtae Kim, and Eun Seon Cho. Constructing reversible li deposition interfaces by tailoring lithiophilic functionalities of a heteroatom-doped graphene interlayer for highly stable li metal anodes. *ACS Applied Materials & Interfaces*, 16(25):32259–32270, 2024.

[11] Ohhyun Kwon, Tae Yong Kim, Taewon Kim, Jihyeon Kang, Seohyeon Jang, Hojong Eom, Seyoung Choi, Junhyeop Shin, Jongkwon Park, Myeong-Lok Seol, et al. Intelligent stress-adaptive binder enabled by shear-thickening property for silicon electrodes of lithium-ion batteries. *Advanced Energy Materials*, 14(20):2304085, 2024.

[12] Haoran Du, Yanhao Dong, Qing-Jie Li, Ruirui Zhao, Xiaoqun Qi, Wang-Hay Kan, Liumin Suo, Long Qie, Ju Li, and Yunhui Huang. A new zinc salt chemistry for aqueous zinc-metal batteries. *Advanced Materials*, 35(25):2210055, 2023.

[13] Terumasa Shimada, Pavel M Usov, Yuki Wada, Hiroyoshi Ohtsu, Taku Watanabe, Kiyohiro Adachi, Daisuke Hashizume, Takaya Matsumoto, and Masaki Kawano. Long time CO2 storage under ambient conditions in isolated voids of a porous coordination network facilitated by the "magic door" mechanism. *Advanced Science*, 11(2):2307417, 2024.

[14] Jinseok Koh, Choah Kwon, Hyunjeong Kim, Eunchong Lee, Akihiko Machida, Yuki Nakahira, Yun Jeong Hwang, Kouji Sakaki, Sangtae Kim, and Eun Seon Cho. Defect-driven evolution of oxo-coordinated cobalt active sites with rapid structural transformation for efficient water oxidation. *ACS nano*, 18(42):28986–28998, 2024.

[15] Yoyo Hinuma. Neural network potential molecular dynamics simulations of (La, Ce, Pr, Nd) 0.95 (Mg, Zn, Pb, Cd, Ca, Sr, Ba) 0.05 F2. 95. *The Journal of Physical Chemistry B*, 128(49): 12171–12178, 2024.

[16] Akira Miura, Koki Muraoka, Kotaro Maki, Saori Kawaguchi, Kazuhiro Hikima, Hiroyuki Muto, Atsunori Matsuda, Ichiro Yamane, Toshihiro Shimada, Hiroaki Ito, et al. Stress-induced martensitic transformation in na3ycl6. *Journal of the American Chemical Society*, 146(36): 25263–25269, 2024.

[17] Kosuke Watanabe, Takuma Higo, Koki Saegusa, Sakura Matsumoto, Hiroshi Sampei, Yuki Isono, Akira Shimojuku, Hideki Furusawa, and Yasushi Sekine. Oxidative dehydrogenation of ethane combined with CO2 splitting via chemical looping on In2O3 modified with Ni–Cu alloy. *ACS Catalysis*, 15(7):5876–5885, 2025.

[18] Tetsuya Honbo, Yutaro Ono, Kota Suetsugu, Mitsuo Hara, Attila Taborosi, Kentaro Aoki, Shusaku Nagano, Michihisa Koyama, and Yuki Nagao. Effects of alkyl side chain length on the structural organization and proton conductivity of sulfonated polyimide thin films. *ACS Applied Polymer Materials*, 6(21):13217–13227, 2024.

[19] Kaoru Hisama, Ksenia V Bets, Nitant Gupta, Ryo Yoshikawa, Yongjia Zheng, Shuhui Wang, Ming Liu, Rong Xiang, Keigo Otsuka, Shohei Chiashi, et al. Molecular dynamics of catalyst-free edge elongation of boron nitride nanotubes coaxially grown on single-walled carbon nanotubes. *ACS nano*, 18(45):31586–31595, 2024.

[20] Han Kim, Taeseok Kim, Hong Keun Chung, Jihoon Jeon, Sung-Chul Kim, Sung Ok Won, Ryosuke Harada, Tomohiro Tsugawa, Sangtae Kim, and Seong Keun Kim. Sustained area-selectivity in atomic layer deposition of ir films: Utilization of dual effects of o3 in deposition and etching. *Small*, 20(46):2402543, 2024.

[21] Seongmin Jin, Choah Kwon, Aram Bugaev, Bartu Karakurt, Yu-Cheng Lin, Louisa Savereide, Liping Zhong, Victor Boureau, Olga Safonova, Sangtae Kim, et al. Atom-by-atom design of cu/zro x clusters on mgo for co2 hydrogenation using liquid-phase atomic layer deposition. *Nature Catalysis*, 7(11):1199–1212, 2024.

[22] Hyesun Kim, HyeonJi Kim, Wonsik Kim, Choah Kwon, Si-Won Jin, Taejun Ha, Jae-Hyeok Shim, Soohyung Park, Aqil Jamal, Sangtae Kim, et al. Facile synthesis of nanoporous mg crystalline structure by organic solvent-based reduction for solid-state hydrogen storage. *Nature Communications*, 15(1):10800, 2024.

[23] Takahiro Ishikawa, Yuta Tanaka, and Shinji Tsuneyuki. Evolutionary search for superconducting phases in the lanthanum-nitrogen-hydrogen system with universal neural network potential. *Physical Review B*, 109(9):094106, 2024.

[24] Jongmin Bae, Choah Kwon, See-On Park, Hakcheon Jeong, Taehoon Park, Taehwan Jang, Yoonho Cho, Sangtae Kim, and Shinhyun Choi. Tunable ion energy barrier modulation through aliovalent halide doping for reliable and dynamic memristive neuromorphic systems. *Science Advances*, 10(23):eadm7221, 2024.

[25] Joe D Morrow and Volker L Deringer. Indirect learning and physically guided validation of interatomic potential models. *The Journal of Chemical Physics*, 157(10), 2022.

[26] Ishan Amin, Sanjeev Raja, and Aditi Krishnapriyan. Towards fast, specialized machine learning force fields: Distilling foundation models via energy hessians. *arXiv preprint arXiv:2501.09009*, 2025.

[27] John LA Gardner, Daniel F Toit, Chiheb Ben Mahmoud, Zoé Faure Beaulieu, Veronika Juraskova, Laura-Bianca Paşca, Louise AM Rosset, Fernanda Duarte, Fausto Martelli, Chris J Pickard, et al. Distillation of atomistic foundation models across architectures and chemical domains. *arXiv preprint arXiv:2506.10956*, 2025.

[28] Duo Zhang, Xinzijian Liu, Xiangyu Zhang, Chengqian Zhang, Chun Cai, Hangrui Bi, Yiming Du, Xuejian Qin, Anyang Peng, Jiameng Huang, et al. DPA-2: a large atomic model as a multi-task learner. *npj Computational Materials*, 10(1):293, 2024.

[29] Filip Ekström Kelvinius, Dimitar Georgiev, Artur Toshev, and Johannes Gasteiger. Accelerating molecular graph neural networks via knowledge distillation. *Advances in Neural Information Processing Systems*, 36:25761–25792, 2023.

[30] Sakib Matin, Alice EA Allen, Emily Shinkle, Aleksandra Pachalieva, Galen T Craven, Benjamin Nebgen, Justin S Smith, Richard Messerly, Ying Wai Li, Sergei Tretiak, et al. Teacher-student training improves accuracy and efficiency of machine learning interatomic potentials. *arXiv preprint arXiv:2502.05379*, 2025.

[31] Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, et al. Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024.

[32] Zhuoran Qiao, Matthew Welborn, Animashree Anandkumar, Frederick R Manby, and Thomas F Miller. OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *The Journal of chemical physics*, 153(12), 2020.

[33] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International conference on machine learning*, pages 9377–9388. PMLR, 2021.

[34] Ralf Drautz. Atomic cluster expansion for accurate and transferable interatomic potentials. *Physical Review B*, 99(1):014104, 2019.

[35] So Takamoto, Satoshi Izumi, and Ju Li. TeaNet: Universal neural network interatomic potential inspired by iterative electronic relaxations. *Computational Materials Science*, 207:111280, 2022.

[36] Ivan S Novikov, Konstantin Gubaev, Evgeny V Podryabinkin, and Alexander V Shapeev. The MLIP package: moment tensor potentials with MPI and active learning. *Machine Learning: Science and Technology*, 2(2):025002, 2020.

[37] Yunxing Zuo, Chi Chen, Xiangguo Li, Zhi Deng, Yiming Chen, Jörg Behler, Gábor Csányi, Alexander V Shapeev, Aidan P Thompson, Mitchell A Wood, et al. Performance and cost assessment of machine learning interatomic potentials. *The Journal of Physical Chemistry A*, 124(4):731–745, 2020.

[38] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[39] Pawel Jozwik, Wojciech Polkowski, and Zbigniew Bojar. Applications of Ni3Al based intermetallic alloys—current stage and potential perceptivities. *Materials*, 8(5):2537–2568, 2015.

[40] Bowen Deng, Yunyeong Choi, Peichen Zhong, Janosh Riebesell, Shashwat Anand, Zhuohan Li, KyuJung Jun, Kristin A Persson, and Gerbrand Ceder. Systematic softening in universal machine learning interatomic potentials. *npj Computational Materials*, 11(1):9, 2025.

[41] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin a. Persson. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013. ISSN 2166532X. doi: 10.1063/1.4812323. URL `http://link.aip.org/link/AMPADS/v1/i1/p011002/s1&Agg=doi`.

[42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[43] Maarten De Jong, Wei Chen, Thomas Angsten, Anubhav Jain, Randy Notestine, Anthony Gamst, Marcel Sluiter, Chaitanya Krishna Ande, Sybrand Van Der Zwaag, Jose J Plata, et al. Charting the complete elastic properties of inorganic crystalline compounds. *Scientific data*, 2 (1):1–13, 2015.

A Broader Impact

LightPFP is a framework designed for efficient training of MLIPs that can support large-scale simulations. By significantly improving data efficiency, computational scalability, and accuracy, LightPFP holds potential to accelerate material discovery. For example, breakthroughs enabled by LightPFP

could contribute to the development of carbon-neutral energy solutions and renewable energy systems, driving sustainable technological progress and improving quality of life globally. However, similarly to any powerful technology, the potential misuse of material discovery technologies bears ethical considerations. The same advancements that enable progress in sustainable technology could be exploited for harmful purposes, such as the creation of weapons or hazardous chemicals. We discourage the use of LightPFP or related technologies for applications that could negatively impact society.

B Additional details of student pre-training

B.1 Model architecture

MTPs employ a mathematically rigorous descriptor system based on invariant moment tensors that encode atomic environments [36]. In MTP, energy can be calculated by the sum of the atomic energy functions of each atom i in the structure: $E = \sum_i V_i$, where $V_i = \sum_\alpha \xi_\alpha B_\alpha(\mathbf{n}_i)$. ξ_α denotes a learnable coefficient of MTP, B_α denotes a basis function and $\mathbf{n}_i$ denotes a set of $\mathbf{r}_{ij}$, a relative coordinate position of atom i to its neighbors. Each basis function B_α comprises of matrix contractions of moment descriptors $M_{\mu,\nu}$, where μ and ν are non-negative integers. The moment descriptor $M_{\mu,\nu}$ for atom i is defined as:

$$M_{\mu,\nu}(\mathbf{n}_i) = \sum_j f_\mu(\mathbf{r}_{ij}) \underbrace{\mathbf{r}_{ij} \otimes \mathbf{r}_{ij} \otimes \cdots \otimes \mathbf{r}_{ij}}_{\nu \text{ times}}$$

where $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ is the relative position vector to neighbor j within cutoff radius R_{cut}, "$\otimes$" denotes a tensor outer product. The function f_μ described a radial part depending on μ is expressed as

$$f_\mu(|r_{ij}|, z_i, z_j) = \sum_{\beta=1}^{N_Q} c^{(\beta)}_{\mu,z_i,z_j} Q^\beta(|r_{ij}|)$$

where $c^{(\beta)}_{\mu,z_i,z_j}$ is a learnable parameter, z indicates the atomic type, the radial function $Q^\beta(|r_{ij}|)$ is the combination of Chebyshev polynomials of the first kind and cutoff function, and N_Q is the number of polynomials.

Moment descriptors are contracted to form rotationally-invariant basis functions $B_\alpha(\mathbf{n}_i)$ that preserve SO(3) symmetry, enabling accurate representation of complex many-body interactions. The formulation of MTP achieves high data efficiency—basis functions span a complete polynomial space while avoiding explicit angular dependence, enabling accurate fits with small training sets [36, 37]. With training data, we fit MTP to learn the parameters $\xi = \{\xi_i, ..., \xi_{n_B}\}$, where n_B is the number of basis, and $\mathbf{c} = \{c^{(\beta)}_{\mu,z_i,z_j}\}$, where the number of coefficients n_c depends on number of f_μ, element pairs including the pair of itself, and N_Q: $n_c = n_{f_\mu} \times n_{\text{elem-pair}} \times N_Q$.

In this study, the pre-trained MTP model employed in this study utilizes 77 basis functions $B_\alpha(\mathbf{n}_i)$. For each element pair, we employ 4 radial part functions f_μ, where each function comprises 16 Chebyshev polynomials Q^β. Consequently, each element pair is characterized by 64 (16×4) parameters. The cutoff distance R_{cut} was set to 6 angstroms.

B.2 Training method

We employed a comprehensive dataset to train PFP, a universal potential-based graph neural networks. This dataset comprises 86 different elements, covering nearly the entire periodic table and encompassing both equilibrium structures and numerous disordered structures that deviate from equilibrium states. The dataset includes not only bulk phases but also complex structures such as surfaces, adsorption configurations, and clusters. This comprehensive coverage is the fundamental reason why PFP exhibits broad applicability across diverse materials simulations. For dataset details, please refer to Takamoto et al. [3].

However, compared to PFP [3], moment tensor potentials (MTPs) are compact models with limited parameters and constrained expressive power, typically applicable only to single materials systems. Consequently, using MTPs to fit all datasets simultaneously presents significant challenges. Therefore,

Task index	Dataset description
1	Single substance structures
2	Bulk structures with two elements
3	Crystal structures from materials project database
4	Structures with random atomic position
5	Bulk structures with defects
6	Single molecules
7	Structures composed by multiple molecules
8	Molecules with element substitution
9	Adsorption structures of surfaces and molecules
10	Random combinations of surfaces and molecules
11	Slabs
12	Clusters

Table 3: Task definitions for Reptile meta learning algorithm based on datasets trained for PFP [3].

our MTP pre-training strategy aims to optimize the model to facilitate subsequent fine-tuning for individual tasks, instead of maximizing accuracy across all datasets. To achieve this objective, we employed the Reptile meta-learning algorithm [38].

The Reptile algorithm operates by iteratively sampling tasks from a task distribution and updating model parameters to enhance the model's ability to rapidly adapt to new tasks. In our implementation, we partitioned the complete dataset into 12 specific tasks based on structural types, as detailed in Table 3. During each inner loop iteration, we select a task (i.e., a dataset containing specific structural types such as single molecules) to train the MTP model. Given the substantial size of each task's dataset, we limit training to one epoch per inner loop before proceeding to the parameter update. The model parameters are then updated according to the following formula:

$$\delta\theta = \theta_i - \theta,$$

$$\theta \leftarrow \theta + \beta\delta\theta,$$

where θ represents the MTP parameters, θ_i denotes the parameters after the i-th inner loop, and β is a hyperparameter in the Reptile algorithm that controls the magnitude of the meta-update step during training. In our implementation, β is set to 0.5. We iteratively repeat the task sampling and inner-loop/meta-update procedures for 100 iterations until convergence of energy, forces, and stress is observed across all datasets.

We employed the Adam optimization method with a learning rate of 1×10^{-3}. The model was trained for 1 epochs with a batch size of 256. Total pre-training time was approximately 100 hours.

B.3 Parameter extraction from pre-trained student

Our pre-trained student model contains 86×86×4×16 training parameters for the radial function c and additional 77 coefficients for the basis functions ξ. The modular structure of MTP enables selective parameter extraction during inference or fine-tuning, significantly enhancing computational efficiency. The extraction procedure is straightforward, depending on elements used for the task. For example, when handling a material containing only H and O elements, we can extract the relevant subset of the radial function parameter tensor—specifically a 2×2×4×16 matrix corresponding to these elements, while maintaining the coefficients of the basis function unchanged. Consequently, although the pre-trained model may contain numerous parameters, it automatically reduces to a compact, element-specific model equivalent in size to those trained from scratch for the particular material system.

C Additional details on Ni_3Al experiments

C.1 Training dataset generation

We use the Ni_3Al crystal structure downloaded from materials project [41] (mp-2593) as an input structure to generate datasets. A full dataset containing 1529 Ni_3Al structures is generated following the static and dynamic sampling methods.

The static methods are compression, deformation of lattices, and displacement of atomic positions. In the compression method, the lattice constants of the input structure are isotropically scaled from 95% to 105% with an interval of 0.5%. Atomic positions are scaled along with the lattice constants. In the deformation method, a supercell consisting $2 \times 2 \times 2$ replication of the unit cell of the input structure are first created. Then, for 6 independent components of strain tensors with strains from -0.02 to 0.02 with an interval of 0.005 are applied to the supercell to generate new structures. In the displacement of atomic positions, we generate 20 structures by randomly picking one atom and displacing it from the equilibrium position along x, y, or z axis by 0.5 angstrom.

The dynamic method uses molecular dynamics (MD) with initial configurations of both defect-free bulk structure and bulk structure with vacancies, as well as surface structures. For defect-free bulk structure (the input structure), we create $2 \times 2 \times 2$ supercells and perform 20000 steps MD simulations with an sampling interval of 100 at the temperatures of 500 K, 1000 K, and 1500 K. The same condition are applied to the MD simulations, however, with 10000 MD steps for $3 \times 3 \times 3$ and with 2000 MD steps for $4 \times 4 \times 4$ supercells. For bulk structures with vacancies, we create three $2 \times 2 \times 2$ supercells and randomly delete one atom. The structures are then subject to MD simulations with 2000 MD steps while other simulation parameters remain the same as above. Similar MD simulations with 1000 MD steps were repeated again for three $3 \times 3 \times 3$ with one vacancy. For surface structures, we generate 6 surface structures by cutting the input structure along (111), (110), (100), (221), (211), and (210) directions. The six surface structures are then used to perform 1000 steps MD simulations with sampling temperature of 300 K, 600 K, 900 K and an sampling interval of 100.

Smaller datasets for testing data efficiency are generated by two methods, subsampling and direct sampling. In the subsampling methods, datasets with sizes of 100, 200, 400, and 800 are generated by randomly sampling from the full dataset with 1529 structures. For each size, five datasets are created for obtaining the uncertainty on testing tasks. In the direct sampling method, dataset generation is performed by applying the previously used static and dynamic sampling approaches, but with the number of MD steps in all MD simulations reduced to fractions of 1/16, 1/8, 1/4, and 1/2 of the original.

C.2 Model training setups

Finetuned student models were trained on the pre-trained model described in Appendix B.2. MTP architecture is identical to that of pre-trained student. Both scratch-trained student models and finetuned student models are training in the same manner.

We split the dataset into 90% for training and 10% for validation, using the validation set both to select the model with the lowest validation loss and to assess validation errors. Optimization was performed using Adam [42] with a batch size of 128, following a three-stage training procedure. n the first stage (125 epochs), the loss coefficients for energy, forces, and stress were set to $(10^{-5}, 10, 10^{-5})$. In the second stage (250 epochs), they were adjusted to $(1, 0.1, 10)$. In the third stage (125 epochs), the loss coefficients were automatically determined to balance the three losses. Specifically, we first computed the total validation loss of the second-stage model using the stage-two coefficients as loss weight. The coefficient for the energy loss was then calculated as the total weighted validation loss divided by three and further divided by the energy loss from stage two. The coefficients for forces and stress were calculated analogously, each using their respective loss from stage two.

The total training spanned 500 epochs. In all stages, mean squared loss was used for energy, force, and stress loss training. A linear warmup learning rate scheduler was applied, increasing the learning rate from zero to its stage-specific maximum during the first 20% of epochs in each stage, and then linearly decaying it to approach zero by the final epoch. The learning rates for stage 1, stage 2, and stage 3 were set to 0.1, 0.01, and 0.01, respectively. Training can be done on a single GPU within half an hour for each setting.

C.3 Property calculation

We calculate errors for energies, forces, stresses, lattice length, elastic tensors, elasticities, phonon frequencies, surface energies, and vacancy formation energies. For clarity, only the errors for energies, forces, phonon frequencies, and surface energies, are present in the main body. All results are shown in Figure 4. The error calculations are detailed below. All of them are difference between the

predictions of student models and those of PFP. The mean absolute errors of energies, forces, and stresses are evaluated on the validation set (10% of input datasets). Lattice length tests are performed through the geometry optimization of the Ni_3Al crystal structure (mp-2593), and the relative errors are averaged over three lattice lengths (a, b, and c). Elastic tensors and elasticity are calculated through the geometry optimization of the crystal structure and the linear fitting between strain and stress. The errors of elastic tensors are averaged over six elastic tensor components. The elasticity errors are the averaged errors over bulk modulus, shear modulus, Young's modulus, and Poisson's ratio. The error of phonon spectra is the average over the errors of phonon energies at each wave vector in phonon spectra. The error of surface energy is the average over the errors of different facets up to largest index of 2. Vacancy formation energies are calculated by systematically removing each atom from the initial Ni_3Al structure and referring to the potential energy of the pristine bulk structure.

C.4 Comparison of pre-trained and scratch-trained student

Due to space constraints, we report only a subset of all properties we evaluated in the main body of the paper. Here, we present all the properties used to compare pre-trained and scratch-trained students in Figure 4. A similar trend is observed across all properties: the pre-trained student consistently achieves superior performance and exhibits greater training stability.

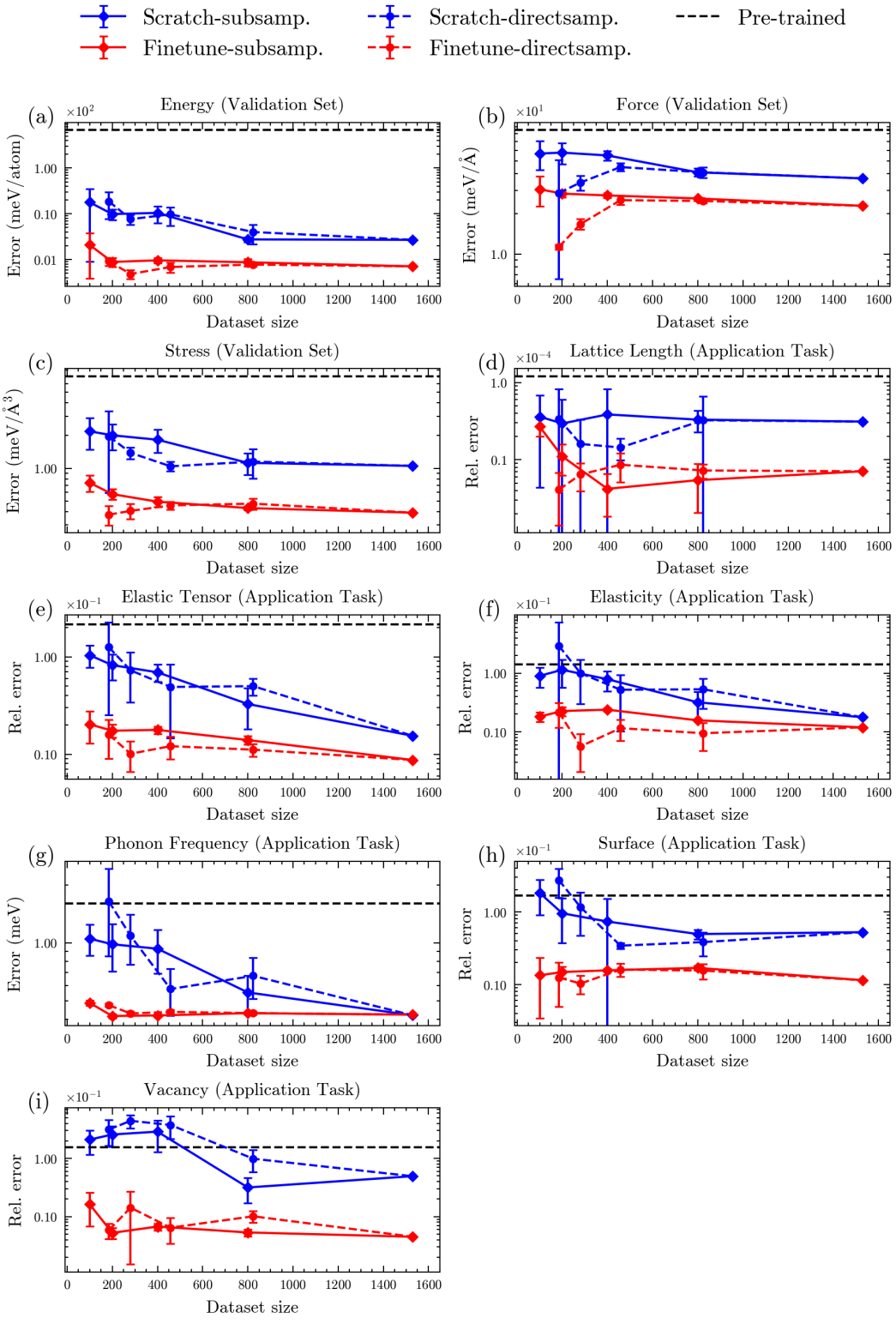

Figure 4: Comparison of data efficiency between finetuned pre-trained and scratch-trained student models.

D Additional details on high entropy alloy experiments

D.1 Training dataset generation

The composition of HEA is 20% each of Al, Co, Cr, Fe, and Ni. HEAs have attracted significant attention due to their exceptional mechanical properties. However, their complex multi-element nature poses challenges for training MLIPs. Datasets are constructed for the training of LightPFP and MTP-DFT. Given that HEAs are solid solutions composed of five elements without an uniform configuration, each lattice site exhibits a highly diverse local atomic environment due to elemental variation. To efficiently sample training data, we adopt a "random substitution" strategy: starting from a pure Al structure, we randomly replace lattice sites with Co, Cr, Fe, or Ni atoms at a 20% probability. Molecular dynamics simulations are then used to sample training structures. This process is repeated many times to diversify the dataset. The initial structures include FCC bulk crystals, slab structures with Miller indices less than 4, and grain boundaries with Σ values less than 10. The composition of the LightPFP dataset is shown in Table 2. Using PFP for data sampling took 24 hours, followed by 1 hour of model training, totaling 25 hours.

The MTP-DFT dataset was generated following the same strategy. PFP-driven molecular dynamics yielded 1012 structures, which were then calculated using VASP to obtain energies, forces, and stresses. Notably, since the trained model is intended for use on surfaces and grain boundaries, the dataset includes corresponding configurations. However, some surfaces and grain boundaries can only be represented by relatively large atomic models even when their sizes are minimized as much as possible, such as the (3 1 1) slab (144 atoms) and the Σ9 38.94°/[1 1 0] (2 -2 -1) grain boundary (140 atoms). These increase the difficulty of DFT calculations, which scale cubically with number of atoms, highlighting the advantage of using universal potentials for sampling. Constructing this DFT dataset took 637 hours on a single GPU. Using ab-initio driven molecular dynamics sampling would have taken an estimated 60,000 hours, making MLIP construction prohibitively expensive. Subsequently, we spent 1 hour training the MTP-DFT model, similarly to LightPFP.

D.2 Property calculation

We used the four MLIP models to compute important properties of the HEA, with DFT results serving as ground truth. We began with the equation of state. Starting from the equilibrium crystal structure, we varied the lattice constant from -5% to +5%, then optimized atomic positions and lattice parameters at fixed volumes. This yielded energy–volume curves, which were fitted using the Birch-Murnaghan equation to extract equilibrium volume and bulk modulus. Calculations were performed on a bulk HEA structure with 256 atoms. We then computed the elastic properties using the stress–strain methodology [43] to obtain the elastic tensor, from which bulk modulus, Young's modulus, and shear modulus were derived. The same bulk HEA structure was used. Surface energies were also evaluated. The surface structure with low Miller index is collected for training. To test extrapolation reliability, we selected surfaces with higher Miller indices for evaluation. The surface formation energy was calculated using the formula:

$$\gamma_{\text{surf}} = \frac{E_{\text{surf}} - \frac{n_{\text{surf}}}{n_{\text{bulk}}} E_{\text{bulk}}}{2A_{\text{surf}}},$$

where E_{surf} is the energy of a slab with two surfaces, E_{bulk} is the energy of the bulk HEA, n_{surf} and n_{bulk} are the atom counts in the surface and bulk structures, and A_{surf} is the surface area. Given the randomness of HEA structures, the energies of surface structures and bulk structure are averaged from 5 different configurations that share the same lattice but different arrangement of elements. Finally, we computed grain boundary formation (GB) energies. The training set included low-mismatch grain boundaries (CSL $\sum < 10$). For testing, we selected several grain boundary structures with $\sum > 10$. The formation energy was calculated using the following equation:

$$\gamma_{\text{GB}} = \frac{E_{\text{GB}} - \frac{n_{\text{GB}}}{n_{\text{bulk}}} E_{\text{bulk}}}{2A_{\text{GB}}},$$

where E_{GB} and E_{bulk} are the energy of GB and bulk structures, n_{GB} and n_{bulk} are their atoms counts, and A_{GB} is the grain boundary area. Again, we averaged over five GB and bulk configurations.

Table 4: Composition of the high entropy alloy dataset.

Type of structure	Sampling method	Number of structures	Number of atoms
LightPFP Dataset (labeled by PFP)			
crystal	substitution+MD	2040	206040
boundary	substitution+MD	6200	1083760
slab	substitution+MD	1398	66816
Total		**9638**	**1356616**
MTP-DFT Dataset (labeled by DFT)			
crystal	substitution+MD	531	42484
boundary	substitution+MD	286	9152
slab	substitution+MD	195	8724
Total		**1012**	**60360**

D.3 HEA dataset statistics

Table 4 shows the statistics of the high entropy alloy dataset. Although it can be seen that PFP-labeled dataset is larger than DFT-labeled dataset, the data collection time of PFP is much faster (24 hours) than DFT (637 hours).

D.4 Full results.

Due to space constraints, we only report some parts of the evaluation for mechanical properties, surface energy, and grain boundary energy in the main body of the paper. Figure 6 illustrates the force error across four different models, while Figure 5 compares inference speed, training time, and the maximum number of atoms supported by various methods. Table 5 presents the complete comparison results of DFT and MLIP models on HEA properties. It is important to note that MTP-DFT and LightPFP share the same model architecture, and thus can support the same maximum number of atoms.

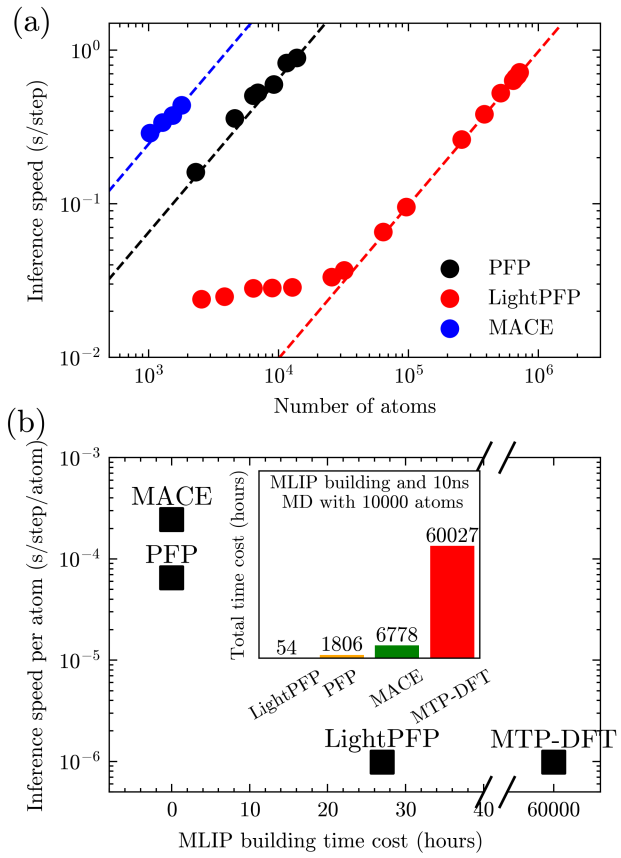

Figure 5: (a) Maximum number of atoms on single GPU with PFP, MACE, LightPFP (MTP-DFT has the same inference speed as LightPFP); (b) Benchmark of inference speed and model building time cost (time spent for training data generation and fine-tuning) of PFP, MACE, LightPFP, and MTP-DFT.

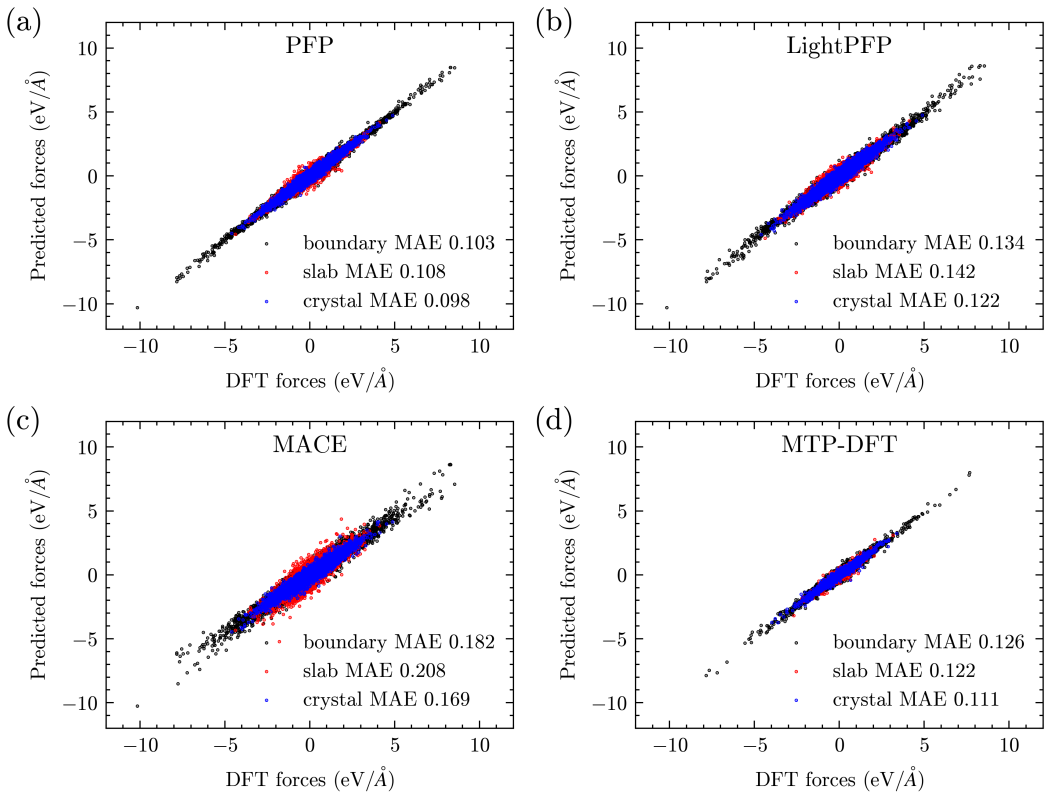

Figure 6: Parity plot of DFT forces and predicted forces by different MLIPs, (a) PFP (b) LightPFP (c) MACE and (d) MTP-DFT.

Table 5: Comparison of DFT and MLIP models on HEA properties

Property	**DFT**	**PFP**	**LightPFP**	**MACE**	**MTP-DFT**
Equation of State					
Volume (Å^3/atom)	11.58	**11.51**	*11.51*	11.48	11.29
Bulk modulus (GPa)	165.64	**165.66**	*164.35*	159.18	162.27
Mechanical Properties (GPa)					
C11	195.2	202.5	196.3	177.2	197.2
C22	211.4	206.9	203.3	183.5	202.7
C33	197.5	206.7	204.3	182.7	203.1
C12	140.9	145.9	151.7	145.9	153.3
C13	142.9	152.9	156.6	148.3	157.6
C23	131.1	137.9	144.9	141.3	148.2
C44	116.5	109.4	106.2	80.2	103.7
C55	124.0	114.2	110.6	84.6	107.1
C66	120.3	112.9	109.9	83.9	106.8
Bulk modulus	159.23	165.45	167.81	157.14	169.02
Shear modulus	69.99	65.79	60.42	45.05	58.54
Young's modulus	183.14	174.27	161.84	123.36	157.44
Average Error	–	**7.20**	*10.65*	23.35	12.55
Surface Energy (eV/Å^2)					
(4, 1, 0)	0.127	0.136	0.133	0.121	0.126
(4, 1, 1)	0.170	0.171	0.165	0.167	0.168
(4, 2, 1)	0.142	0.149	0.148	0.134	0.145
(4, 3, 0)	0.139	0.144	0.143	0.137	0.143
(4, 3, 2)	0.137	0.143	0.145	0.126	0.142
(4, 4, 1)	0.148	0.153	0.153	0.146	0.154
(4, 4, 3)	0.171	0.178	0.175	0.174	0.176
Average Error	–	0.0058	0.0053	*0.0052*	**0.0036**
Grain Boundary Energy (eV/Å^2)					
Σ13 22.62/[1 0 0]	0.0559	0.0621	0.0578	0.0424	0.0523
Σ15 48.19/[1 2 0]	0.0794	0.0809	0.0787	0.0602	0.0825
Σ13 147.80/[1 1 1]	0.0378	0.0300	0.0294	0.0206	0.0268
Σ13 67.38/[1 0 0]	0.0584	0.0617	0.0563	0.0332	0.0504
Σ11 129.52/[1 1 0]	0.0955	0.0735	0.0771	0.0670	0.0737
Average Error	–	*0.0081*	**0.0063**	0.0207	0.0095

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe how our method works and provide experimental results that LightPFP demonstrates good accuracy and efficiency.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We described limitation in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provided information in the main body and appendix for reproducing our results with Matlantis [4].

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not have a plan to release the code at this moment because of the usage of proprietary software Matlantis.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Described in experiment sections and Appendices C and D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide error bars in the data efficiency experiments, which shows that pre-training the students can be superior. Howver, we did not give error bar in s

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Described in Section 3.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Experiments do not involve human subjects or participants. We also discussed Broader impact in Appendix A.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Described in Appendix A

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.