Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2025, **4**, 2253

MatterTune: an integrated, user-friendly platform for fine-tuning atomistic foundation models to accelerate materials simulation and discovery†

Lingyu Kong,[a] Nima Shoghi,[a] Guoxiang Hu,[c] Pan Li[b] and Victor Fung*[a]

Received 17th April 2025
Accepted 11th July 2025

DOI: 10.1039/d5dd00154d

rsc.li/digitaldiscovery

Geometric machine learning models such as graph neural networks have achieved remarkable success in recent years in chemical and materials science research for applications such as high-throughput virtual screening and atomistic simulations. The success of these models can be attributed to their ability to effectively learn latent representations of atomic structures directly from the training data. Conversely, this also results in high data requirements for these models, hindering their application to problems which are data sparse which are common in this domain. To address this limitation, there is a growing development in the area of pre-trained machine learning models which have learned general, fundamental, geometric relationships in atomistic data, and which can then be fine-tuned to much smaller application-specific datasets. In particular, models which are pre-trained on diverse, large-scale atomistic datasets have shown impressive generalizability and flexibility to downstream applications, and are increasingly referred to as atomistic foundation models. To leverage the untapped potential of these foundation models, we introduce MatterTune, a modular and extensible framework that provides advanced fine-tuning capabilities and seamless integration of atomistic foundation models into downstream materials informatics and simulation workflows, thereby lowering the barriers to adoption and facilitating diverse applications in materials science. In its current state, MatterTune supports a number of state-of-the-art foundation models such as ORB, MatterSim, JMP, MACE, and EquformerV2, and hosts a wide range of features including a modular and flexible design, distributed and customizable fine-tuning, broad support for downstream informatics tasks, and more.

1 Introduction

Geometric machine learning models, such as graph neural networks (GNNs), have had a revolutionary impact on machine learning for the chemical and materials science domains. These models represent a paradigm shift away from the extensive and often time-consuming feature engineering required in traditional informatics approaches[1–3] and toward data-driven representation learning, thereby enabling them to be broadly applicable to a wide range of applications ranging from chemical property prediction and screening, molecular dynamics simulations, to the inverse design of new materials, and more. This has led to, by all accounts, an explosive growth in recent years of studies utilizing GNNs for these aforementioned tasks trained on existing materials datasets.[4–9]

Almost all GNN models of this class operate on the general principle of taking the atomic identity and structure of a molecule or crystal as inputs, and mapping this geometric information to their corresponding property labels. In the case of GNNs, this information is encoded in the node and edge attributes of a graph, which are then processed through message passing operations to yield latent atom-level and system-level embeddings or representations. From this embedding, the property labels can then be obtained *via* non-message passing layers, commonly referred to as a readout function or an output head. Starting from seminal examples such as SchNet[10] and CGCNN,[11] subsequent models have incorporated increasingly sophisticated advancements including the incorporation of many-body interactions,[12–14] equivariant features,[15–17] and transformer-like architectures,[18,19] though nearly all still follow the same aforementioned general principles. Although these GNNs have become increasingly accurate and scalable with these improvements, they are inherently data driven and invariably function poorly for instances where training data is sparse. This limitation prevents their widespread application to the majority of materials science-related problems where data may range in the hundreds or even fewer samples.

[a] *School of Computational Science and Engineering, Georgia Institute of Technology, USA. E-mail: victorfung@gatech.edu*

[b] *School of Electrical and Computer Engineering, Georgia Institute of Technology, USA*

[c] *School of Materials Science and Engineering, Georgia Institute of Technology, USA*

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d5dd00154d

A rapidly growing area of research towards greater data efficiency of GNNs is in the pre-training of GNNs. This approach generally involves first training these models on large upstream datasets (the "pre-training" stage) before continuing the training on the smaller downstream dataset(s) of interest (the "fine-tuning" stage). This process enables the models to learn robust, transferable representations without requiring the final property labels. Two general strategies exist for pre-training: supervised and unsupervised. In supervised pre-training, GNNs are initially trained on certain explicit property labels which are sufficiently generalizable to downstream needs. Properties such as energies, forces, and sometimes stresses from quantum mechanical calculations were found to be particularly effective for pre-training,[20–22] among others.[23,24] In unsupervised pre-training, unlabeled data are used instead, and the model is then trained on objectives such as a contrastive loss or denoising loss.[25–28] While pre-training can be applied to datasets of any size and complexity, including artificial ones, there is a growing effort to pre-train GNNs on datasets which attempt to cover the full range of the chemical and materials space. Once pre-trained, these models should, in theory, be generalizable to downstream datasets of arbitrary complexity and properties. These models we term as "atomistic foundation models (FMs)" (Table 1). A growing numbers of studies have shown atomistic FMs can improve accuracies of GNNs significantly over models trained from scratch (*i.e.* without pre-training), as well as reduce data requirements by an order of magnitude or more.[20–22]

Here, it is important to note the parallel development of universal interatomic potentials (UIPs), which are models trained to be broadly applicable force fields for systems of arbitrary complexity on compositions across the periodic table.[12,21,28–32] Whereas UIPs are intended to be used out-of-the-box for one specific task (as force fields), pre-trained models require an additional fine-tuning step before they can be used, but are applicable to tasks beyond force fields. Nevertheless, the distinction between UIPs and pre-trained models can become blurred as in some cases, the training procedures and datasets for UIPs can be identical to those used in the creation of pre-trained atomistic models, namely when the pre-training objective is on energies and forces. Consequently, one can note that while not all pre-trained models can serve as UIPs, in general most UIPs should serve as capable pre-trained models.

Despite the demonstrated potential of atomistic FMs, general adoption by the broader scientific community is currently lacking, in large part due to the limitations of the available software infrastructure for its usage. While there is, to date, ample infrastructure for UIPs, this does not extend to any tasks beyond being used as force fields, such as materials property prediction. There is also limited standardization across different UIPs and atomistic FMs, resulting in a different package being needed for each different model, hampering benchmarking and workflow development. Finally, there is limited support for the customizability of the fine-tuning procedure, which is often hard-coded as a black-box method. As such, these existing packages do not currently fulfill the role of servicing atomistic FMs for general-purpose usage.

To address these limitations, we developed a modular, integrated, and user-friendly framework, called MatterTune, for fine-tuning atomistic FMs to be applied to a broad range of materials science applications. The development of MatterTune follows several general design principles:

(1) Highly generalizable and flexible abstractions that enable systematic extension while enforcing the necessary standardization.

(2) Modular framework decoupling models, data, algorithms, and applications, enabling a high degree of adaptability and customizability for different materials informatics tasks.

(3) Intuitive and user-friendly interfaces that simplify model fine-tuning and their application to downstream tasks.

So far, MatterTune has integrated several open-source atomistic FMs including JMP,[20] ORB,[28] EquiformerV2,[18] MatterSim-V1,[21] and MACE.[33] We fine-tuned these models using the MatterTune platform and evaluated them on representative materials informatics tasks, including molecular dynamics simulations, property screening, and materials discovery, demonstrating the performance and reliability of the MatterTune platform and its capabilities for data-efficient learning.

2 Methods

The MatterTune framework primarily consists of four components: a model subsystem, a data subsystem, a trainer subsystem, and an application subsystem, each covering a core component of a fine-tuning task. In addition, we have

Table 1 Overview of some recently released atomistic FMs

Model	Release year	Num. params	Dataset size	Training obj.
MACE-MP-0	2023	4.69M	1.58M	Energy, forces, stress
GNoME	2023	16.2M	16.2M	Energy, forces
MACE-MPA-0	2024	9.06M	12M	Energy, forces, stress
MatterSim-v1	2024	4.55M	17M	Energy, forces, stress
ORB-v1	2024	25.2M	32.1M	Denoising + energy, forces, stress
JMP-S	2024	30M	120M	Energy, forces
JMP-L	2024	235M	120M	Energy, forces
EqV2-S	2024	31.2M	1.58M	Energy, forces, stress
EqV2-M	2024	86.6M	102M	Energy, forces, stress
DPA3-v1-MPtrj	2025	3.37M	1.58M	Energy, forces
DPA3-v1-OpenLAM	2025	8.18M	143M	Energy, forces

abstracted the key components that require standardization and extensibility, making it straightforward for MatterTune to quickly support new models and new data formats. The MatterTune package can be accessed at **https://github.com/Fung-Lab/MatterTune**.

2.1 Flexible and generalizable abstractions

In order to support various kinds of current and future atomistic FMs for the myriad of applicable materials systems and applications, some generalizable abstractions for atomistic FMs are needed. Several considerations need to be addressed here: first, atomistic FMs can employ diverse architectural paradigms, ranging from graph neural networks to transformers. Second, each atomistic FM has its own supported input format, internal representations, and computational requirements, all of which must be supported and standardized. Finally, the implementation should cover the breadth of possible materials informatics tasks and handle heterogeneous property types ranging from scalar quantities to vector fields. Considering these factors, MatterTune's architecture is centered around three key abstractions to enable systematic extension while enforcing the necessary standardization:

- Data abstraction: the purpose of data abstraction is to provide unified support for as many input formats as possible for training and inference. We develop a minimal data abstraction that defines a dataset $\mathcal{D}$ as a mapping $f: \mathbb{N} \rightarrow \mathcal{S}$, where $\mathcal{S}$ represents the space of atomic structures in a standardized format. Given that different atomistic FMs require varying input formats, we choose `ase.Atoms` in ASE package[34] as the standardized format of $\mathcal{S}$. Individual atomistic FMs can then implement the necessary transformations from `ase.Atoms` to their respective input formats. Since the `ase.Atoms` format can store all structural and label information needed for training and prediction, this abstraction is broadly applicable.

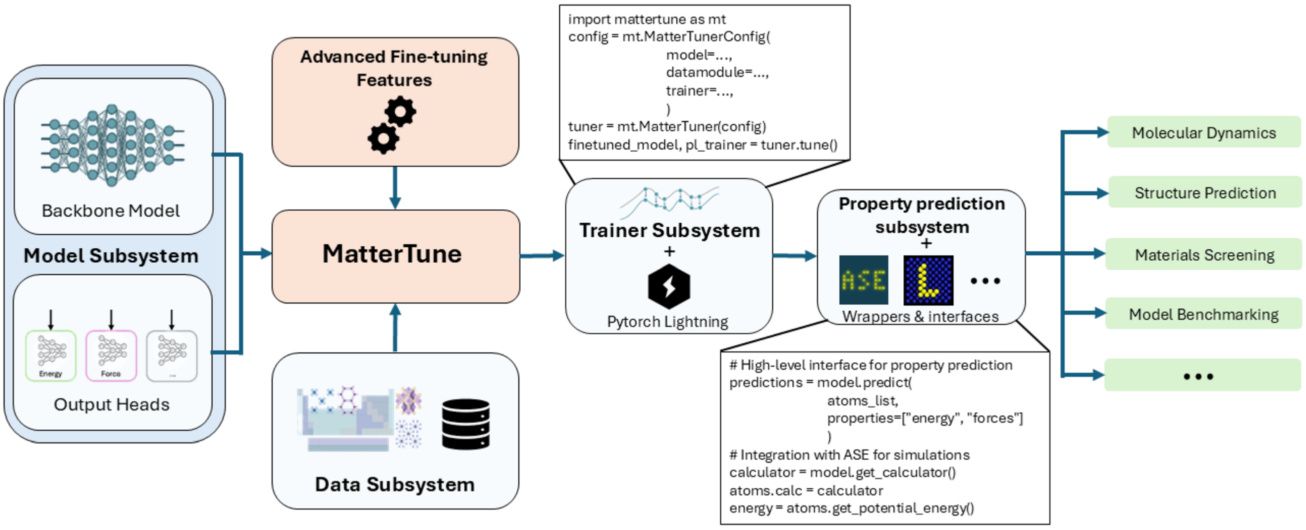
- Property abstraction: we introduce a property schema system that formally separates the specification of physical properties from their model implementation, allowing users to focus solely on the types of properties they require from the model without concerning themselves with the details of how these properties are realized in FMs. This separation also enables the framework to handle both established properties like energy and forces as well as custom properties defined by users, and enforces type safety and physical constraints (*e.g.*, energy conservation) in a property-specific manner.

- Backbone abstraction: the purpose of backbone abstraction is to provide a set of unified functional interfaces for using different backbones, regardless of various FMs' completely different model architectures. For example, some key functions include the `model_forward` function, which handles forward propagation during prediction, and the `atoms_to_data` function, which converts input structures from the `ase.Atoms` format into the format required by the model. This abstraction ensures simplicity and consistency in model usage while enabling each model to retain its native internal representations and implementations.

2.2 A modular and standardized framework design

As illustrated in Fig. 1, the modular framework enables MatterTune to decouple data, models, training algorithms, and downstream applications, allowing users to freely select and combine these components. This approach distinguishes MatterTune from other frameworks that rely solely on direct API calls. Following the aforementioned core abstractions, the framework is organized into several distinct subsystems:

- The data subsystem follows the aforementioned data abstraction and handles conversion between various materials science formats and a universal internal representation used by the MatterTune framework. Currently we have provided built-in support for common formats like XYZ, JSON, and ASE



```
import mattertune as mt
config = mt.MatterTunerConfig(
                    model=...,
                    datamodule=...,
                    trainer=...,
                    )
tuner = mt.MatterTuner(config)
finetuned_model, pl_trainer = tuner.tune()
```

```
# High-level interface for property prediction
predictions = model.predict(
                    atoms_list,
                    properties=["energy", "forces"]
                    )
# Integration with ASE for simulations
calculator = model.get_calculator()
atoms.calc = calculator
energy = atoms.get_potential_energy()
```

Fig. 1 Overview of the MatterTune framework.

databases, which can be readily expanded to include additional formats as needed.

- The model subsystem is designed around the backbone and property abstractions, allowing users to simply specify the type of atomistic FM and the desired properties to predict in order to declare and construct a model. All implementation details—such as loading checkpoints, constructing output heads, handling input data, and performing forward passes, are automatically managed by MatterTune, respecting the original implementation of each atomistic FM. This approach enables users to leverage atomistic FMs without requiring in-depth knowledge of their underlying architecture and implementation.

- The trainer subsystem handles the general training, validation, and checkpointing of FMs. A key design choice is made to integrate the training subsystem with PyTorch Lightning,[35] a widely used and feature-rich training platform. This enables a range of critical capabilities while maintaining a clean separation of concerns between the model implementation and the training process. The integration of Lightning's abstractions allows MatterTune to maintain a modular and extensible architecture while still providing a simple, high-level interface for end users. Currently, MatterTune provides support for various optimizers and learning rate schedulers on the Lightning platform. It also includes implementations of data preprocessing statistics, exponential moving average, and other fine-tuning techniques, allowing users to select them freely. In addition, Lightning's callback features allow for ample flexibility for implementing more advanced fine-tuning strategies.

- The property prediction subsystem provides the means for users to access the trained FMs in an easy-to-use and intuitive manner to enable to quick integration of downstream materials tasks. This is accomplished by providing implementations of flexible wrapper classes for both general and targeted use-cases without having to deal with model architecture complexities or Lightning internals. As a starting point, we have implemented an `MatterTuneCalculator` which heritages the ASE[34] calculator interface, enabling direct use with established molecular dynamics and structure optimization algorithms available within the ASE package. For high-throughput material property prediction tasks, we have designed the `MatterTunePropertyPredictor` as a wrapper around atomistic FMs, enabling batch prediction in parallel. We are working on implementing interfaces for additional materials informatics simulation and computation software, such as LAMMPS.

2.3 A unified fine-tuning technique tool-kit

In the process of reproducing the fine-tuning experimental results for atomistic FMs, we discovered that reproducing their reported performance in the literature heavily depends on the specific fine-tuning approaches and techniques employed, and that different models require distinct fine-tuning settings. This observation motivated us to develop a unified fine-tuning toolkit on the MatterTune platform. Currently, our fine-tuning toolkit meets the requirement of ensuring that the integrated models can largely reproduce their expected fine-tuning performance as shown in Section 3. To this end, we have implemented the following:

- A variety of optimizers and learning rate schedulers. So far, MatterTune has supported the Adam,[36] AdamW,[37] and SGD optimizers, as well as various learning rate schedulers including linear, step, exponential, cosine, and reduce-on-plateau. In addition to these, MatterTune also enables customization of optimizers and learning rate schedulers based on user needs. We support the application of different learning rates to different parts of the model, a technique that has been used and shown to be beneficial in the fine-tuning of models of the JMP series. Furthermore, we support combining multiple learning rate schedulers to achieve more sophisticated dynamic adjustments, such as cosine warm-up.

- Training generalization techniques such as Exponential Moving Average (EMA). Although the ablation studies in ref. 20 suggest that EMA does not significantly improve fine-tuning performance on datasets such as MD17: aspirin, MD22: stachyose, QM9: $\Delta\varepsilon$, MatBench: MP E form, QMOF: band gap, and SPICE: solvated amino acids, we believe these datasets are still not small enough in scale. In our experiments described in Section 3, where models are fine-tuned using only 30 data points, we observed that EMA actually helps improve both the stability and performance of fine-tuning.

- A comprehensive normalization system that handles both standard statistical normalization and physics-informed normalization schemes. Fine-tuning of FMs may involve multiple targets—for example, training a force field model typically involves three targets: energy, forces, and stress. Proper normalization helps balance the loss scales of these targets, ensuring that the training process converges more smoothly without being dominated by any single target. MatterTune currently supports not only standard normalization methods such as mean-std and root-mean-square, but also composition-based normalization using element-wise regression. Additionally, the normalization system is designed to be composable, allowing multiple normalization schemes to be applied in sequence.

3 Results

In the following experiments, we will demonstrate the performance of fine-tuned atomistic FMs on a variety of representative tasks and benchmarks using the MatterTune platform. These tasks include molecular dynamics simulations, materials property prediction, and materials discovery. The goal of these experiments is to showcase the correctness of MatterTune's implementation as well as its flexibility across diverse downstream applications.

We first note that MatterTune maintains strict adherence to the original implementations of each integrated atomistic FM. However, many models do not provide openly available details on fine-tuning parameters and techniques for specific tasks. Given that hyperparameter tuning is both complex and computationally expensive, we did not perform exhaustive hyperparameter optimization for the benchmarks shown below. As a result, we cannot guarantee that each atomistic FM achieves its best possible performance on these tasks. Nonetheless, for tasks with publicly accessible reference results, we have made a dedicated effort to reproduce them.

3.1 Property prediction on MatBench and high-throughput screening on novel out-of-distribution materials

Atomistic FMs should be broadly applicable to various chemical and materials systems, and can be effectively fine-tuned as capable property predictors. This makes atomistic FMs highly promising for high-throughput property screening. In MatterTune, we provide support for constructing direct property prediction output heads for atomistic FMs, even if they are not present in their original implementations. To showcase this capability, we selected several FMs for fine-tuning on multiple tasks from Matbench (v0.6),[38] a well-established materials informatics benchmark.

The performance of JMP-S, ORB-V2, and Equiformer-31M-mp fine-tuned on Matbench is shown in Table 2. In our current tests, we perform fine-tuning on fold 0 of each dataset. In the table, we also list the fine-tuning performance on Matbench from the original JMP-S paper, as well as the best performance on the Matbench leaderboards. It should be noted that, since model fine-tuning can be a delicate process, variations in fine-tuning methods and hyperparameter configurations can lead to significant differences in the results. In our experiments, all models across all tasks were fine-tuned using the same configuration, so we cannot guarantee that the results reported on MatterTune represent the optimal performance of the models. Nonetheless, by comparing the fine-tuning results of JMP-S on MatterTune with those reported in the original paper, we found that we reproduced the reported accuracy in most tasks, with the only exception being formation energy, where our fine-tuning result was inferior to the original. Moreover, out of the three fine-tuned models JMP-S, ORB-V2, and Equiformer-31M-mp, the best model in each task significantly outperforms the current leading models trained from scratch on Matbench leaderboard.

Although Matbench provides a train-test split for evaluating fine-tuned models, they are drawn from the same original dataset distribution, which prevents them from accurately reflecting the models' performance on unseen new materials. To address this, we further performed high-throughput property predictions on approximately 404 763 new structures provided by the GNoME dataset (release 2024-11-21)[39] which are distinct from the original Materials Project dataset. For demonstrative purposes, we also screened out structures with band gaps between 1 eV and 3 eV and compared the classification performance with the ground truth. The results are shown in Table 3. The results indicate that the ORB-V2 model, which achieved the highest test accuracy on the band gap task in Matbench, also delivered the best performance in band gap property screening on the GNoME dataset.

Table 3 High-throughput screening on GNoME band gap data

	JMP-S	ORB-V2	Equiformer-31M-mp
MAE (eV)	0.052	0.039	0.044
Accuracy (%)	98.16	98.80	98.53
Recall (%)	86.25	90.33	89.92
$F1$	0.826	0.884	0.861

3.2 Few-shot fine-tuning and molecular dynamics simulations for liquid water

The original MatterSim paper presents a compelling demonstration of the capability of atomistic FMs to achieve reliable accuracy on specific tasks through fine-tuning with minimal data. In their experiments on a liquid water system, the authors compared three scenarios: zero-shot performance, fine-tuning using the full training set (900 samples), and fine-tuning with only a small subset (30 samples). The results showed that even just 30 samples for fine-tuning, the model achieved roughly the same level of accuracy as using 900 samples, and in both cases the models could accurately reproduce the radial distribution functions (RDFs) when compared with the experimental data.

To demonstrate the few-shot capability of the atomistic FMs in general, we followed the same experimental setup described in the original MatterSim paper. Out of the entire 1000 available ambient water data,[40–42] we uniformly sampled 100 structures based on the energy distribution as a validation set and used the rest as the 900-sample dataset. We then randomly selected 30 structures from the 900-sample dataset and subsequently repeated them until a new dataset comprising 900 samples was obtained. We refer to this dataset as the 30-sample dataset. We fine-tuned various FMs on both the 900-sample and the 30-sample dataset and evaluated models' mean absolute errors on the validation set. The results are shown in Table 4.

Table 2 Evaluation of property prediction performance of various models on Matbench

Task (units)	Best on leaderboards (mean)	JMP-S-baseline (fold0)	JMP-S (fold0)	ORB-V2 (fold0)	EqV2-31M-mp (fold0)
Dielectric (unitless)	0.271	0.133	0.146	0.142	**0.111**
JDFT2D (meV per atom)	33.19	20.72	**19.42**	21.44	23.45
Log GVRH ($\log_{10}$ (GPA))	0.067	0.06	0.059	**0.053**	0.056
Log KVRH ($\log_{10}$ (GPA))	0.049	0.044	**0.033**	0.046	0.046
MP E_form (meV per atom)	17.0	13.6	25.2	**9.4**	24.5
MP gap (eV)	0.156	0.119	0.119	**0.093**	0.098
Perovskites (eV per unitcell)	0.027	0.029	0.029	0.033	**0.026**

Table 4 Fine-tuning performance of various FMs on ambient water dataset

	MatterSim V1-1M		JMP-S		ORB-V3 Omat-conserv.		EqV2-31M mp		MACE-MP-0a medium	
	900-Sample	30-Sample	900-Sample	30-Sample	900-Sample	30-Sample	900-Sample	30-Sample	900-Sample	30-Sample
MAE_E (meV per atom)	1.21	1.20	3.06	5.65	2.50	1.15	2.76	4.98	1.19	3.01
MAE_F (meV $Å^{-1}$)	38.37	40.65	**19.98**	30.17	33.73	34.04	22.41	35.21	48.65	51.24

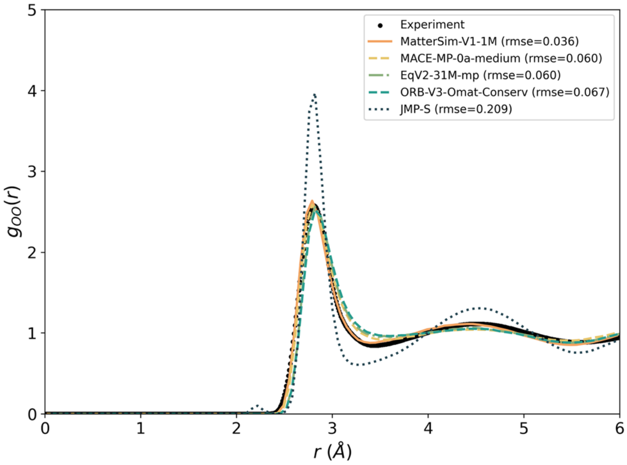


Fig. 2 Oxygen–oxygen radial distribution functions of ambient water under 298 K obtained from foundation model based MD simulations. Black dots represent experimental references from ref. 44 and 45. All models shown are versions fine-tuned on a 30-sample dataset. The legend also reports the root-mean-square error between each model's MD-derived RDF curve and the experimental data.

We further conducted 200 ps molecular dynamics (MD) simulations of a water structure with 192 atoms per unit cell at 298 K using the FMs fine-tuned on the dataset of 30 samples shown above. The MD thermostat engine employs the NPT ensemble implemented in ASE (without external stress to keep the cell fixed). The results of the radial distribution function analysis are shown in Fig. 2. Interestingly, we observed that although all five models performed well in terms of MAE, as shown in Table 4, the results of MD simulation varied significantly. MatterSim-V1-1M fits the experimental data best, and the results from MACE-MP-0a-medium, EquiformerV2-31M-mp, and ORB-V3-Omat-Conserv remain broadly acceptable, though show some discrepancies. In contrast, the JMP-S model, which has the lowest force MAE on the validation set, produces the RDF curve with the largest deviation. This observation echoes the statement in ref. 43, which cautions that evaluating models solely based on force MAE can lead to misleading conclusions. One possible explanation is that MatterSim-V1-1M, MACE-MP-0a-medium, and ORB-V3-Omat-Conserv are energy-conserving force-field models, whereas JMP-S employs direct force prediction, which makes its MD simulations less stable. However, EquiformerV2-31M-mp also uses direct force prediction, yet still yields reasonably accurate RDF results.

During these MD simulations of the ambient-water system, we assessed whether the FMs integrated and trained within MatterTune incur any systematic runtime overhead in either training or prediction relative to their original implementations. The results show that no systematic overhead is brought by MatterTune. Full numerical details are provided in Section S2 of the ESI.†

3.3 Zero-shot prediction and structural geometry optimization

Novel material discovery and structure prediction are among the central challenges in the computational materials sciences. Matbench Discovery[46] provides a benchmark for evaluating models in accurately determining stable materials structures. We validated MatterTune's implementation of model loading and zero-shot prediction, as well as its correct support for geometry optimization, by reproducing the performance of several models on Matbench Discovery. We employed the ASE-implemented FIRE optimizer and the ExpCellFilter unit cell

Table 5 Zero-shot performance of various models on Matbench-discovery[a]

	EqV2 S DeNS baseline	EqV2 S DeNS	MatterSimV1 5M baseline	MatterSimV1 5M	ORB-V2 baseline	ORB-V2
$F1$ ↑	0.815	0.792	0.862	0.842	0.880	0.866
DAF ↑	5.042	4.718	5.852	5.255	6.041	5.395
Prec ↑	0.771	0.756	0.895	0.876	0.924	0.899
Acc ↑	0.941	0.925	0.959	0.949	0.965	0.957
MAE ↓	0.036	0.035	0.024	0.024	0.028	0.027
R^2 ↑	0.788	0.780	0.863	0.848	0.824	0.817

[a] Metric definitions: all metrics listed in the table follow the same definitions as their counterparts on the Materials Discovery leaderboard; upward (↑) and downward (↓) arrows denote that higher or lower values are preferred, respectively. $F1$: harmonic mean of precision and recall for stable/unstable materials classification; DAF: discovery acceleration factor measuring how much better the models classify thermodynamics stability compared to random guessing; Prec: precision of classifying thermodynamics stability; Acc – accuracy of classifying thermodynamics stability; MAE – mean absolute error of predicted *vs.* DFT convex hull distance; R^2 – coefficient of determination.

filter for all models, using 0.02 eV Å^{-1} and a maximum of 500 steps as the cut-off conditions for structural relaxation. The final results, along with a comparison to the leaderboard outcomes, are presented in Table 5. The results indicate that the reproduced outcomes are within an acceptable error range relative to the leaderboard results. The very minor discrepancies

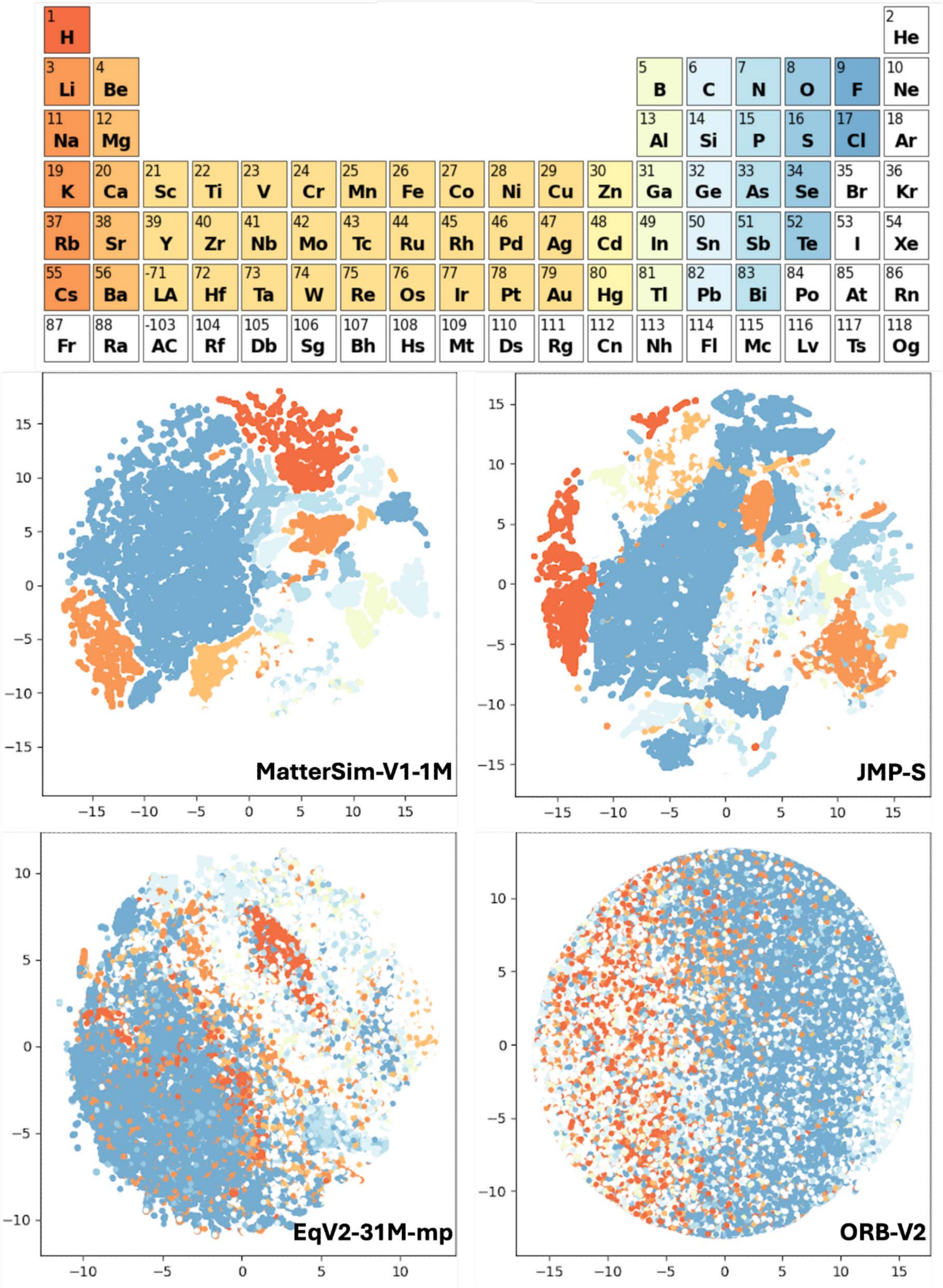


Fig. 3 TSNE visualization of four pre-trained FMs' node representations on a subset of MPTraj dataset containing 5000 structures. Node representations are colored by element types, with similar colors assigned to elements in the same group.

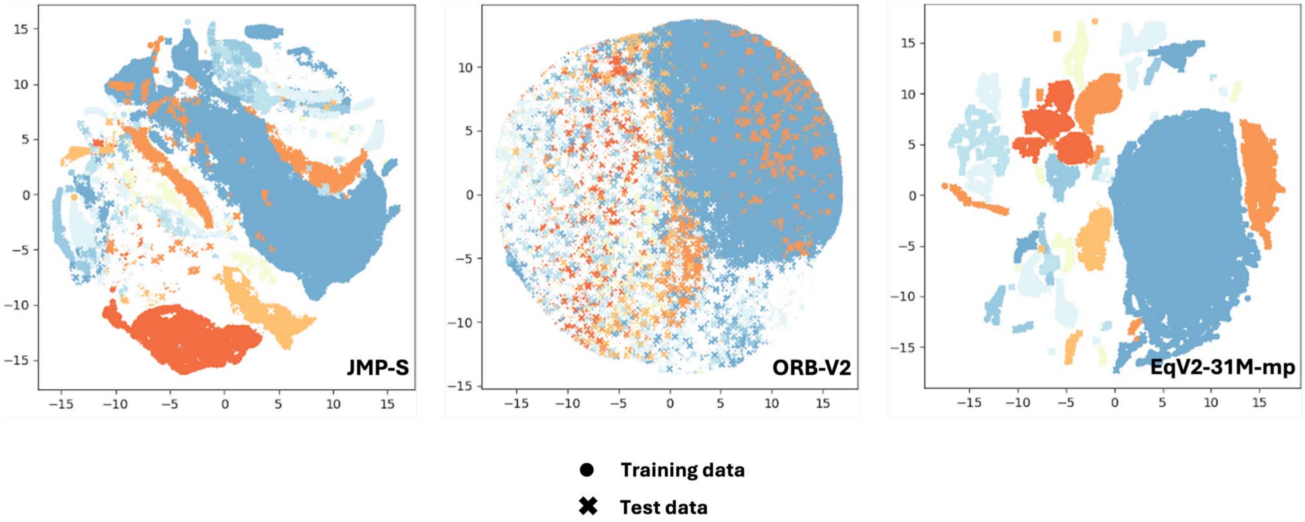


Fig. 4 TSNE visualization of node representations on MP_E_form from four FMs fine-tuned on the same dataset. The ● symbols represent structures from the training set, while the × symbols indicate structures from the test set.

observed may stem from the choice of optimizer, the unit cell filter, and numerical precision, among other factors.

3.4 Representation space visualization

It is generally believed that one of the main reasons FMs excel in downstream tasks is their ability to learn high-quality, general-purpose geometric representations during pre-training. We leveraged MatterTune's internal feature extraction capabilities to export the node representations for four different models. To demonstrate this, we selected two datasets: MPTraj and MP_E_form, and visualize the representations generated by pre-trained FMs and those fine-tuned for specific tasks. For the MatterSim, JMP, and ORB models, which intrinsically contain node feature vectors, we used these as the structural representations of atomic local environments. In Equiformer, where node features are divided into multiple irreducible representation channels, we extracted the invariant (*i.e.*, $l = 0$, $m = 0$) features as a representative structural descriptor. Extracting the latent representations of atomistic FMs can provide a window into interpreting their performance, as well as be used for purposes such as active learning.[47]

In Fig. 3, the representations of four different atomistic FMs are visualized using the t-SNE algorithm. The results show that the MatterSim and JMP models clearly capture the clustering of elements within the same group. This is to be expected from a chemical perspective, and suggests some level of transferable chemical knowledge is trained into these models. In contrast, the clustering for Equiformer and ORB models is less pronounced, especially for ORB. These results highlight the remarkable diversity in the internal representations of current atomistic FMs, which may arise due to differences in training objectives, training data, and model architectures.

We further visualized the representation spaces of the fine-tuned models (Fig. 4). We selected the MP_E_form dataset motivated by the fact that the fine-tuning results of the three models on this dataset showed notable differences (as detailed in Table 2). The visualization results reveal apparent similarities between the fine-tuned JMP-S and Equiformer models which cluster around element type, whereas ORB has no clear clustering similar to the non fine-tuned case. This pattern is consistent with JMP-S and Equiformer having similar MP_E_form accuracies, while ORB is significantly lower. However, the underlying link between the differences in the representation and the fine-tuning performance is still unclear and deserves further investigation, which can be easily facilitated with MatterTune.

4 Discussion and conclusions

The MatterTune offers a flexible, generalizable framework that seamlessly integrated multiple atomistic FMs and supports tasks such as molecular dynamics simulations, materials property predictions, and materials discovery. MatterTune offers users a wide range of choices in data formats, model architectures, and training configurations. As a consequence of the modular design of MatterTune, users can freely mix and match these components according to their performance needs and specific requirements. Our experimental results to replicate the models' original reported performance on ambient water systems and JMP's Matbench experiments demonstrate that this unified approach to integrating atomistic FMs is both feasible and does not compromise model performance. We note that MatterTune is in active development and additional features will be planned in the future. The integration of additional newly developed atomistic FMs, as well as interfaces to more materials simulation software such as LAMMPs into MatterTune is ongoing. Furthermore, advanced fine-tuning procedures commonly seen in other contexts such as large language models will also be explored and implemented within the MatterTune toolkit.

To summarize, MatterTune is an effort to standardize and unify atomistic FMs while providing user-friendly interfaces for fine-tuning and applications. MatterTune also serves as a playground for experimenting and applying advanced fine-tuning algorithms to atomistic FMs. By lowering the barrier to the use of atomistic FMs, we aim to make them broadly applicable across a wide range of materials science challenges, especially in materials simulations and informatics. Furthermore, we hope that the MatterTune platform can provide a foundation for exploring how to fine-tune atomistic FMs more effectively to meet the increasingly demanding requirements of materials science research.

Data availability

The MatterTune platform, together with all the code and data used for the experiments in this paper are available on Github at **https://github.com/Fung-Lab/MatterTune**, and can be found at this DOI: **https://doi.org/10.5281/zenodo.15859271**.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 using NERSC award BES-ERCAP0032102.

References

1 R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi and C. Kim, Machine learning in materials informatics: recent applications and prospects, *npj Comput. Mater.*, 2017, **3**(1), 54.

2 F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi and M. Ceriotti, Physics-inspired structural representations for molecules and materials, *Chem. Rev.*, 2021, **121**(16), 9759–9815.

3 J. Damewood, J. Karaguesian, J. R. Lunger, A. R. Tan, M. Xie, J. Peng, *et al.*, Representations of materials for machine learning, *Annu. Rev. Mater. Res.*, 2023, **53**(1), 399–426.

4 V. Fung, J. Zhang, E. Juarez and B. G. Sumpter, Benchmarking graph neural networks for materials chemistry, *npj Comput. Mater.*, 2021, **7**(1), 84.

5 E. Chien, M. Li, A. Aportela, K. Ding, S. Jia, S. Maji, *et al.*, Opportunities and challenges of graph neural networks in electrical engineering, *Nat. Rev. Electr. Eng.*, 2024, **1**(8), 529–546.

6 P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, *et al.*, Graph neural networks for materials science and chemistry, *Commun. Mater.*, 2022, **3**(1), 93.

7 Z. Jiao, Y. Liu and Z. Wang, Application of graph neural network in computational heterogeneous catalysis, *J. Chem. Phys.*, 2024, **161**(17), 107001.

8 Y. Wang, Z. Li and A. Barati Farimani. Graph neural networks for molecules, In *Machine learning in molecular sciences*, Springer, 2023, pp. 21–66.

9 O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, *et al.*, A compact review of molecular property prediction with graph neural networks, *Drug Discovery Today: Technol.*, 2020, **37**, 1–12.

10 K. Schütt, P. J. Kindermans, H. E. Sauceda Felix, S. Chmiela, A. Tkatchenko and K. R. Müller, Schnet: A continuous-filter convolutional neural network for modeling quantum interactions, *Adv. Neural Inf. Process. Syst.*, 2017, **30**, 992–1002.

11 T. Xie and J. C. Grossman, Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties, *Phys. Rev. Lett.*, 2018, **120**(14), 145301.

12 C. Chen and S. P. Ong, A universal graph deep learning interatomic potential for the periodic table, *Nat. Comput. Sci.*, 2022, **2**(11), 718–728.

13 J. Gasteiger, F. Becker and S. Günnemann, Gemnet: Universal directional graph neural networks for molecules, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 6790–6802.

14 K. Choudhary and B. DeCost, Atomistic line graph neural network for improved materials property predictions, *npj Comput. Mater.*, 2021, **7**(1), 185.

15 S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, *et al.*, E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, *Nat. Commun.*, 2022, **13**(1), 2453.

16 S. Doerr, M. Majewski, A. Pérez, A. Kramer, C. Clementi, F. Noe, *et al.*, TorchMD: A deep learning framework for molecular simulations, *J. Chem. Theory Comput.*, 2021, **17**(4), 2355–2363.

17 K. T. Schütt, O. T. Unke and M. Gastegger, Equivariant message passing for the prediction of tensorial properties and molecular spectra, *arXiv*, 2021, preprint, arXiv:2102.03150, DOI: **10.48550/arXiv.2102.03150**, **https://arxivorg/abs/210203150**.

18 Y. L. Liao, B. Wood, A. Das and T. Smidt, Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations, *arXiv*, 2023, preprint, arXiv:230612059, DOI: **10.48550/arXiv.2306.12059**.

19 C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, *et al.*, Do transformers really perform badly for graph representation?, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 28877–28888.

20 N. Shoghi, A. Kolluru, J. R. Kitchin, Z. W. Ulissi, C. L. Zitnick and B. M. Wood, From Molecules to Materials: Pre-training Large Generalizable Models for Atomic Property Prediction, in *International Conference on Learning Representations*, 2024.

21 H. Yang, C. Hu, Y. Zhou, X. Liu, Y. Shi, J. Li, *et al.*, Mattersim: A deep learning atomistic model across elements,

temperatures and pressures, *arXiv*, 2024, preprint, arXiv:240504967, DOI: 10.48550/arXiv.2405.04967.

22 S. Jia, A. R. Parthasarathy, R. Feng, G. Cong, C. Zhang and V. Fung, Derivative-based pre-training of graph neural networks for materials property predictions, *Digital Discovery*, 2024, **3**(3), 586–593.

23 S. Jia, S. Govil, M. Ramprasad and V. Fung, Pre-training Graph Neural Networks with Structural Fingerprints for Materials Discovery. *arXiv*, 2025, preprint, arXiv:250301227, DOI: 10.48550/arXiv.2503.01227.

24 I. Amin, S. Raja and A. Krishnapriyan, Towards Fast, Specialized Machine Learning Force Fields: Distilling Foundation Models via Energy Hessians, *arXiv*, 2025, preprint, arXiv:250109009, DOI: 10.48550/arXiv.2501.09009.

25 Y. Wang, J. Wang, Z. Cao and A. Barati Farimani, Molecular contrastive learning of representations via graph neural networks, *Nat. Mach. Intell.*, 2022, **4**(3), 279–287.

26 T. Koker, K. Quigley, W. Spaeth, N. C. Frey and L. Li, Graph contrastive learning for materials, *arXiv*, 2022, preprint, arXiv:221113408, DOI: 10.48550/arXiv.2211.13408.

27 S. Zaidi, M. Schaarschmidt, J. Martens, H. Kim, Y. W. Teh, A. Sanchez-Gonzalez, *et al.*, Pre-training via denoising for molecular property prediction, *arXiv*, 2022, preprint, arXiv:220600133, DOI: 10.48550/arXiv.2206.00133.

28 M. Neumann, J. Gin, B. Rhodes, S. Bennett, Z. Li, H. Choubisa, *et al.*, Orb: A fast, scalable neural network potential, *arXiv*, 2024, preprint, arXiv:241022570, DOI: 10.48550/arXiv.2410.22570.

29 I. Batatia, D. P. Kovacs, G. Simm, C. Ortner and G. Csányi, MACE: Higher order equivariant message passing neural networks for fast and accurate force fields, *Adv. Neural Inf. Process. Syst.*, 2022, **35**, 11423–11436.

30 B. Deng, P. Zhong, K. Jun, J. Riebesell, K. Han, C. J. Bartel, *et al.*, CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling, *Nat. Mach. Intell.*, 2023, **5**(9), 1031–1041.

31 D. Zhang, X. Liu, X. Zhang, C. Zhang, C. Cai, H. Bi, *et al.*, DPA-2: a large atomic model as a multi-task learner, *npj Comput. Mater.*, 2024, **10**(1), 293.

32 P. M. Lupo, J. Y. Choi, K. Mehta, P. Zhang, D. Rogers, J. Bae, *et al.*, Scalable training of trustworthy and energy-efficient predictive graph foundation models for atomistic materials modeling: a case study with HydraGNN, *J. Supercomput.*, 2025, **81**(4), 618.

33 I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács, J. Riebesell, *et al.*, A foundation model for atomistic materials chemistry, *arXiv*, 2023, preprint, arXiv:240100096, DOI: 10.48550/arXiv.2401.00096.

34 A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, *et al.*, The atomic simulation environment—a Python library for working with atoms, *J. Phys.: Condens. Matter*, 2017, **29**(27), 273002.

35 Pytorch lightning, GitHub, 2019.

36 D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv*, 2014, preprint, arXiv:14126980, DOI: 10.48550/arXiv.1412.6980.

37 I. Loshchilov, F. Hutter, *et al.*, Fixing weight decay regularization in adam, *arXiv*, 2017, preprint, arXiv:171105101, DOI: 10.48550/arXiv.1711.05101.

38 A. Dunn, Q. Wang, A. Ganose, D. Dopp and A. Jain, Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm, *npj Comput. Mater.*, 2020, **6**(1), 138.

39 A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon and E. D. Cubuk, Scaling deep learning for materials discovery, *Nature*, 2023, 80–85.

40 B. Cheng, E. A. Engel, J. Behler, C. Dellago and M. Ceriotti, Ab initio thermodynamics of liquid and solid water, *Proc. Natl. Acad. Sci. U. S. A.*, 2019, **116**(4), 1110–1115.

41 B. Monserrat, J. G. Brandenburg, E. A. Engel and B. Cheng, Liquid water contains the building blocks of diverse ice phases, *Nat. Commun.*, 2020, **11**(1), 5757.

42 Z. Chen, M. L. Berrens, K. T. Chan, Z. Fan and D. Donadio, Thermodynamics of water and ice from a fast and scalable first-principles neuroevolution potential, *J. Chem. Eng. Data*, 2023, **69**(1), 128–140.

43 X. Fu, Z. Wu, W. Wang, T. Xie, S. Keten, R. Gomez-Bombarelli, *et al.*, Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations, *arXiv*, 2022, preprint, arXiv:221007237, DOI: 10.48550/arXiv.2210.07237.

44 L. B. Skinner, C. Benmore, J. C. Neuefeind and J. B. Parise, The structure of water around the compressibility minimum, *J. Chem. Phys.*, 2014, **141**(21), 214507.

45 W. Chen, F. Ambrosio, G. Miceli and A. Pasquarello, Ab initio electronic structure of liquid water, *Phys. Rev. Lett.*, 2016, **117**(18), 186401.

46 J. Riebesell, R. E. Goodall, P. Benner, Y. Chiang, B. Deng, A. A. Lee, *et al.*, Matbench Discovery–A framework to evaluate machine learning crystal stability predictions, *arXiv*, 2023, preprint, arXiv:230814920, DOI: 10.48550/arXiv.2308.14920.

47 J. Qi, T. W. Ko, B. C. Wood, T. A. Pham and S. P. Ong, Robust training of machine learning interatomic potentials with dimensionality reduction and stratified sampling, *npj Comput. Mater.*, 2024, **10**(1), 43.