Force Field Optimization by End-to-End Differentiable Atomistic Simulation

Abhijeet S. Gangan
Department of Civil and Environmental Engineering
University of California, Los Angeles
CA 90095, USA
abhijeetgangan@g.ucla.edu

Ekin Dogus Cubuk
Google DeepMind
Mountain View
CA, USA
cubuk@google.com

Samuel S. Schoenholz
OpenAI, San Francisco
CA, USA

Mathieu Bauchy
Department of Civil and Environmental Engineering
University of California, Los Angeles
CA 90095, USA
bauchy@ucla.edu

N. M. Anoop Krishnan
Department of Civil Engineering,
Yardi School of Artificial Intelligence,
Indian Institute of Technology Delhi,
New Delhi, India 110016
krishnan@iitd.ac.in

Abstract

The accuracy of atomistic simulations critically depends on the precision of force fields, yet traditional, numerical methods often struggle to optimize the potential parameters for reproducing target properties. While recent approaches rely on training these force fields based on first-principle simulations, this alone cannot capture complex material responses such as vibrational, or elastic properties. To this extent, we introduce a framework, employing inner loop simulations and outer loop optimization, that exploits automatic differentiation for both property prediction and force-field optimization by computing gradients of the simulation analytically. We demonstrate the approach by optimizing classical Stillinger-Weber and EDIP potentials for silicon systems to reproduce the elastic constants, vibrational density of states, and phonon dispersion. We also demonstrate how a machine-learned potential can be fine-tuned using automatic differentiation to reproduce any target property such as radial distribution functions. Interestingly, the resulting force field exhibits significantly improved accuracy and generalizability to unseen temperatures than those fine-tuned on energy and force. Finally, we demonstrate the extension of the approach to optimize the force fields towards multiple target properties. Altogether, differentiable simulation, through the analytical computation of gradients of the simulation, offers a powerful tool for both theoretical exploration and practical applications toward understanding physical systems and materials.

1 Introduction

Atomistic simulations offer direct insight into atomic movements, which, in turn, dictate the macroscopic properties of materials [Frenkel and Smit, 2023]. The force field, responsible for predicting the total energy of a system based on atomic coordinates, serves as the cornerstone of these simulations,

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

determining their accuracy and reliability [Torrens, 2012]. The primary objective of a force field is to replicate atomic motions observed in first-principle simulations as closely as possible while faithfully reproducing experimentally observed properties. Traditionally, force field development has focused on fitting parameters to match experimental properties, such as elastic moduli, lattice parameters of crystalline systems, density, and other complex characteristics [Torrens, 2012, Pedone et al., 2022, Purja Pun and Mishin, 2009, Wang et al., 2018, Röcken and Zavadlav, 2024, Carrer et al., 2024, Thürlemann and Riniker, 2023]. However, with the emergence of machine-learned force fields (MLFFs), recent efforts have shifted toward training models directly on energy and force data derived from first-principle computations [Noé et al., 2020, Deringer et al., 2019] or learning them from the observables [Thaler and Zavadlav, 2021, Bishnoi et al., 2023, King et al., 2024b, Zills et al., 2024]. Both approaches share the ultimate goal of optimizing force field parameters by minimizing the discrepancy between simulations and "ground truth"—target properties for classical force fields and energy and force values for MLFFs. Thus, the success of these methodologies critically hinges on the precise computation of the gradients with respect to the simulated quantities.

Efficient and accurate calculation of derivatives is crucial in atomistic simulations, whether for force field optimization or for computing gradient-based properties integral to these simulations [Schoenholz and Cubuk, 2020, Brenner and King, 2023]. In force field optimization, the derivatives of properties with model parameters are essential. However, this task becomes particularly challenging as the number of model parameters increases. One common approach to approximate derivatives is the finite difference method. While straightforward, this method becomes computationally prohibitive in high-dimensional spaces, as the number of function evaluations scales with the dimension. Additionally, the finite difference method is susceptible to truncation and round-off errors, making it impractical for large-scale applications.

Alternatively, symbolic differentiation can be employed [Dürrbaum et al., 2002], where derivatives are computed analytically using computer algebra systems such as Mathematica and SymPy [Meurer et al., 2017]. Although this approach yields precise analytical derivatives, the resulting expressions can grow exponentially complex due to the tree-like structure of operations and the repeated application of the chain rule. Manual differentiation, where gradients are computed by hand and then coded, is another option; however, this method is often time-consuming and prone to human error [Strikwerda, 2004]. Despite these challenges, traditional atomistic simulation packages, such as LAMMPS [Thompson et al., 2022], often rely on manually coded derivatives (forces), which underscores the importance of reliable and efficient derivative computation in these simulations.

The machine learning community has long utilized efficient algorithms such as back-propagation to compute gradients, a technique that is a specialized form of a broader method known as Automated or Algorithmic Differentiation (AD). AD enables the efficient computation of analytical gradients, which is critical for optimizing complex models. Libraries such as JAX and PyTorch have harnessed the power of AD to create end-to-end differentiable simulation codes, which are increasingly being applied to the simulation of physical systems [Müller and Cusdin, 2005, Thangamuthu et al., 2022, Bishnoi et al., 2024, Wang et al., 2023a] and a wide range of optimization and control problems [Tamayo-Mendoza et al., 2018, Thomas et al., 2010, Hu et al., 2019, Chen et al., 2022, Zu and Goodrich, 2024]. On an atomistic simulations front, JAX-MD is an actively maintained package that leverages these capabilities. There have also been other attempts to optimize geometries [Axelrod et al., 2022], control [Wang et al., 2020], sampling [Schwalbe-Koda et al., 2021], and learning pair-potentials [Wang et al., 2023b]. Despite the wider adoption of AD for several optimization problems, there have been limited efforts to exploit its use for direct optimization of force fields for any targeted property of interest [Brenner and King, 2023].

Here, we apply AD as implemented in JAX-MD to optimize force fields for various phases of silicon, aiming to closely match the static and dynamic properties predicted by first-principles simulations such as elastic tensors, vibrational density of states, and radial distribution functions (RDF). We show that AD can be used to identify the optimal parameters for both classical and MLFFs in just 4-5 iterations. Further, we also show that the approach could be easily extended to multi-objective optimizations to tune force fields towards replicating multiple target properties. Our results suggest AD has the potential to revolutionize atomistic simulations by enabling the computation of the gradients of simulation—a feature that can have implications beyond force field optimization.

2 Methodology

2.1 Dataset generation

To demonstrate the capabilities of differentiable atomistic simulation for force-field optimization, we rely on first-principles calculations to generate the ground truth. Without the loss of generality, we focus on silicon (Si), a material that has been extensively studied in the past few decades Sastry and Austen Angell [2003], exhibiting both amorphous and crystalline forms. Several structural, physical, and mechanical properties, such as elastic tensor, phonon density of states, and radial distribution functions, of the material, are computed using first-principles calculation—the framework of plane-wave density functional theory (DFT) as implemented in the Vienna *Ab Initio* Simulation Package (VASP) [Kresse and Hafner, 1993, Kresse and Furthmüller, 1996] is used. The details of VASP settings to generate different properties like the elastic constants, force constants, and RDF with the choice of pseudo-potential (PSP) are mentioned the the Supplementary material A. For the training MLFFs, we employ an existing dataset [Cubuk et al., 2017] that provides the energy and forces of Si for different configurations generated using DFT.

2.2 Differentiable simulation

Atomistic simulations are carried out using the end-to-end differentiable package, namely, JAX-MD. JAX-MD allows computing analytical gradients to model parameters that are not available in traditional MD packages like LAMMPS. To the best of our knowledge, JAX-MD is the only package that allows differentiable simulation of atomistic systems. We do, however, use LAMMPS to check the consistency of the classical simulations. The parameters of the simulation like the timestep, temperature, etc are kept similar to those in the *ab initio* simulation mentioned above to avoid inconsistencies during the simulation. Additional details on how several properties such as force constants are computed are discussed in the Supplementary materials A.

2.3 Training details

For the `scipy` optimization routines the max iteration is set to 200 and the convergence tolerance is made tighter. Other settings are kept as default. For the MPGNN model, a cutoff of 3 Å was used to generate graphs from the atomic coordinates. The GNN is repeated for $k = 2$ messages and the MLP for the update, message functions, and the decoder 64 neurons with 2 layers is used. For training the GNN we used the Adam optimizer with a learning rate of $1e - 4$. The fine-tuning process was carried out for a total of 2000 epochs with a batch size of 64. For the evaluation, we save the model with the lowest validation error.

Loss functions govern the nature of the model training and the final optimized solutions. For the tensor property of rank two, such as the elastic constants C_{ij} and force constants H_{ij}, Frobenius norm is used as the loss function given by:

$$L_F = \sqrt{\sum_i \sum_j |Y_{ij} - \hat{Y}_{ij}|^2}. \tag{1}$$

where, Y_{ij} is true value of the property, with the hat (ˆ) representing the predicted counterpart. For the multi-objective optimization, we combine the Frobenius norm for two properties and use the weights w_1 and w_2 which determine the scaling of the loss values and are hyperparameters.

$$L_T = w_1 \sqrt{\sum_i \sum_j |Y_{ij} - \hat{Y}_{ij}|^2} + w_2 \sqrt{\sum_i \sum_j |Z_{ij} - \hat{Z}_{ij}|^2} \tag{2}$$

For fine-tuning the message passing graph neural network (MPGNN), we use the mean-squared error (6) between energies E and forces $\vec{F}$ for different configurations $i \in$ I.

$$L = \frac{1}{n} \sum_{i=1}^{n} (E_i - \hat{E}_i)^2 + \frac{1}{n} \sum_{i=1}^{n} (\vec{F}_i - \hat{\vec{F}}_i)^2 \tag{3}$$

For finetuning the MPGNN on radial distribution functions, we add a term χ^2 in the loss function to measure the difference between the $g(r)$. We add a hyperparameter w for scaling the χ^2, which is set

to a value of 2000 in the present work.

$$L = \frac{1}{n}\sum_{i=1}^{n}(E_i - \hat{E}_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(\vec{F}_i - \hat{\vec{F}}_i)^2 + w\chi^2 \tag{4}$$

where χ^2 is defined as:

$$\chi^2 = \frac{\sum_i^n [g(r) - \hat{g}(r)]^2}{\sum_i^n [\hat{g}(r)]^2} \tag{5}$$

The performance of the model is evaluated on the root mean squared error (RMSE), L^1 (mean absolute error, MAE)and L^2 (mean squared error, MSE) norms of the predicted quantity such as energy, force or radial distribution function as follows. Note that this metric is primarily used for the MPGNN models.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}; \quad L^1 = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|; \quad L^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

2.4 Interaction potentials for Silicon

We use three different models i.e. the Stillinger-Weber (SW), Environment-dependent interatomic potential (EDIP), and Message Passing Graph neural network (MPGNN) for modeling Silicon. All three models are implemented in the JAX-MD package. The details of the models and their parameters can be found in the supplementary materials A.

2.5 Molecular simulation setup

The elastic constants, C_{ijkl} (see Eq.7), are computed as the double derivative of the energy with strains using the `elasticity.py` module in JAX-MD. For computing the force constants, the Hessian, H_{ij} (see Eq.7), is computed using the auto-grad in JAX and then post-processed by the`phonopy` package. The details of the computation are described in the Supplementary A along with the computation of thermodynamics properties.

$$C_{ijkl} = \frac{1}{V}\frac{\partial^2 U}{\partial \epsilon_{ij}\partial \epsilon_{kl}}; \quad H_{ij} = \frac{\partial^2 U}{\partial R_i \partial R_j} \tag{7}$$

For simplicity, we use the Voigt notation for representing the elastic tensors in the rest of the manuscript with reduced subscripts as C_{ij}. For the molecular dynamics simulation for computing the RDF and training the GNN model, we use both the NVE and NVT ensembles with a time step of 1 fs and with a total of 200 steps with a temperature of 2000 K. The gradient is then computed by differentiating through this simulation. In the testing part, this simulation is repeated for 5000 steps. The results presented are for the NVE ensemble as they are almost identical to the NVT results but are faster in terms of simulation time. All the simulations use periodic boundary conditions.

3 Results

3.1 Differentiable simulation

First, we briefly discuss how the differentiable atomistic simulation framework enables the optimization of the force-field parameters. Fig. 1 shows the overall pipeline employed in the present work. The objective of the optimization problem is to obtain the optimal force-field parameters $\mathbf{\Theta}^*$ associated with the potential energy function $\mathbf{U}(\mathbf{R}, \mathbf{\Theta})$ by minimizing the function f, defined as:

$$\mathbf{\Theta}^* = \text{argmin } f(\mathbf{R}, \mathbf{\Theta}) \tag{8}$$

where, $\mathbf{R} \in \mathbb{R}^{N\times d}$ represents the position vectors associated with N atoms in d dimensions, $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}, \theta_k \in \mathbb{R}$ are the parameters of the force-field, and f is the loss function representing the difference between the ground truth and the predictions from the simulation. Note that $\mathbf{U}$ represents the total energy of the system and the atom-level forces are obtained as the gradient of $\mathbf{U}$. It is worth noting that the loss function is on the properties that can be obtained through simulations only. For instance, despite having accurate interatomic potentials, material properties

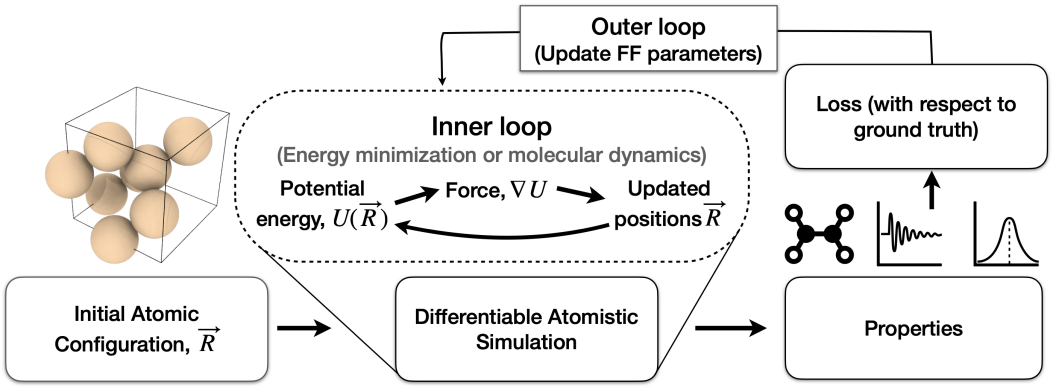

Figure 1: Schematic illustrating differentiable simulation for the optimization problem.

Table 1: Original and optimized parameters for SW and EDIP force fields by minimizing the error on elastic constant using different optimization algorithms.

Force field params	SW		EDIP	
	ϵ (eV)	λ	A (eV)	λ (eV)
Original	2.16826	21.0	7.9821730	1.4533108
BFGS	1.935160956627	33.786738332455	7.191596385156	1.457774753403
CG	1.935160954811	33.786738320322	7.191596392526	1.457774758363
NM	1.935159113520	33.786797615075	7.191619415626	1.457752548290

such as elastic modulus, and phonon density of states can be computed only by performing an atomistic simulation. Here, we optimize the parameters of the force field for a given target property using a differentiable simulation.

Specifically, the framework consists of an inner simulation loop and an outer optimization loop (see Fig. 1). The inner simulation loop carries out traditional atomistic simulation (energy minimization or molecular dynamics) to compute the property. Then, the error between the property computed through the inner simulation loop and the ground truth is computed. Following this, an outer back-propagation loop updates the parameters of the force field to minimize the error. Note that the parameter update is performed by computing the direction in which loss decreases through gradient descent. Thus, differentiable simulation allows the computation of the gradients of a simulation directly with the potential parameters or other input parameters. This approach enables one to bypass the expensive numerical differentiation approaches such as finite difference which demands a large number of data points while still having numerical approximation errors.

3.2 Elastic constants

Now, we demonstrate how differentiable simulation can be employed to optimize the force-field parameters, both classical and machine-learned, towards a target property. We start with the optimization of the classical SW and EDIP potential parameters for Si to match the elastic constants from DFT. Details of the potentials and elastic constants' computation from DFT are provided in the Methodology section. Since the elastic tensor has multiple components, we use the Frobenius norm as the loss function to minimize (1) where Y_{ij} and $\hat{Y}_{ij}$ are matrix elements of C_{ij} computed from forcefields in JAX-MD and DFT, respectively. Note that in this case, the elastic tensor Eq. 7 computation in the inner loop and the backpropagation to minimize the Frobenius norm in the outer loop is performed using automatic differentiation. Thus, in addition to the optimization, the elastic tensor is computed by computing the Hessian of the energy analytically with strains using automatic differentiation. This approach reduces the computational cost associated with multiple simulation steps and avoids any numerical errors associated with the finite difference method.

Table 2: Comparison of elastic constants C_{ij} for original and optimized parameters of SW and EDIP potential along with DFT values.

Force field		C_{11} (GPa)	C_{12} (GPa)	C_{44} (GPa)
SW	Original	151.4131	76.4154	112.8927
	Optimized	162.3052	54.6155	140.8832
EDIP	Original	172.0389	64.6745	145.5683
	Optimized	162.3106	54.6135	140.8778
DFT		153.2141	56.7969	149.6653

Table 3: Original and optimized parameters for SW and EDIP potential from force constant matching.

Force field params	SW		EDIP	
	ϵ (eV)	λ	A (eV)	λ (eV)
Original	2.16826	21.0	7.9821730	1.4533108
Optimized - CG	1.576510867700	22.759369692763	5.710778625018	0.746479371508

To optimize the force-field parameters, we consider three different optimization algorithms, that is, Conjugate gradient (CG), Broyden–Fletcher–Goldfarb–Shanno (BFGS), and Nelder–Mead (NM). CG and BFGS are gradient-based optimization algorithms that can utilize gradients computed from JAX-MD directly, whereas NM is a gradient-free minimizer. For each minimizer, optimization is performed with four different initial conditions to obtain the statistics. Although there are several parameters for these potentials, without the loss of generality, we choose ϵ and λ for SW and A and λ for EDIP as they govern the scaling of the two and three-body terms, respectively Justo et al. [1998], Stillinger and Weber [1985]. The other parameters are kept fixed and the default values of the potentials are given in table 7. The approach could be used to optimize other parameters also by simply computing their gradients as well.

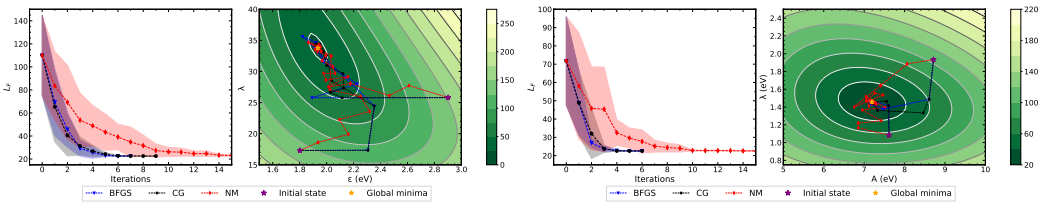

Figure 2: The learning curve and the optimization pathway in the parameter space with different optimizers for SW (top) and EDIP (bottom) potentials. The contour values are the values of the loss function at each parameter.

Figures 2(a) and (b) show the loss curve for the SW and EDIP potentials. We observe that the gradient-based algorithms (BFGS and CG) saturate faster than NM for both SW and EDIP. Further, both BFGS and CG exhibit similar behavior for both potentials with loss saturating at 6 and 4 iterations for SW and EDIP potential, respectively. The optimized parameters for SW and EDIP are shown in Table 1. We observe that the optimized values of parameters obtained by all the minimizers are comparable while being significantly different from the original values. To evaluate this behavior further, we plot the loss landscape for the potential parameters and project the minimization trajectory onto the landscape. Figures 2(c) and (d) show the loss landscape for SW and EDIP potentials, respectively, for two random initial states. We observe that the loss landscape exhibits a convex nature. Both BFGS and CG reach the global minima in 4 steps irrespective of the initial condition, consistent with the loss curve. Thus, the differentiable can find the global minimum for the optimization in an accelerated fashion.

We also evaluate the C_{ij} values obtained from the optimized parameters for SW and EDIP potentials (see Table 2). For SW Si, we observe that while C_{12} and C_{44} values are closer to DFT than the original potential, C_{11} values are worse. This reveals an inherent limitation of the SW potential wherein optimizing all the C_{ij} parameters together leads to a competing effect. Thus, the optimization performed using the Frobenius norm, which minimizes the error of the sum of all the C_{ij} parameters, results in a Pareto optimal solution. For EDIP Si, similar results are observed where C_{11} and C_{12} values are closer to the DFT solution than the original parameters, while C_{44} is poorer. Interestingly, we observe that the final C_{ij} values obtained by employing the optimized parameters of both EDIP and SW potentials yield similar values. This further confirms the competing nature of C_{ij} values for both the potentials that are being minimized by the Frobenius norm.

3.3 Vibrational properties

Now, we focus on the vibrational properties computed from the force constants, such as the phonon dispersion and the vibrational density of states. Note that the force constant is computed as the Hessian of the energy. Thus, in this case, we optimize the force-field parameters of SW Si and EDIP Si potentials to minimize the Frobenius norm between the force constants computed using JAX-MD and DFT. Thus, in this case, the inner loop consists of the computation of the force constants and the outer loop consists of backpropagation to minimize the loss function by optimizing the force-field parameters. Thus, in the inner loop Hessian of the energy is computed with respect to the Cartesian coordinates analytically using automatic differentiation to compute the force constants. In the outer

loop, to minimize the loss (Eq. 1) between the computed Hessian and ground truth obtained from DFT, analytical gradients of the Hessian H_{ij} with respect to the model parameters are computed using JAX-MD. This gradients are used to update the force-field parameters iteratively. Since we observed that both BFGS and CG yielded similar results in the case of elastic tensor, here, we employ only the CG optimizer to minimize the loss function.

Figures 5(a) and (b) show the loss curves associated with the outer loop iterations for minimizing the error on force constants for SW and EDIP force-field parameters, respectively. Similar to the elastic constant computation, we observe that the loss saturates in 6 and 3 steps for SW and EDIP potentials, respectively. To understand this behavior further, we visualize the loss landscape with respect to the force-field parameters (see Figures 5(c) and (d)). Similar to the elastic constant, the optimization problem is convex in nature as evidenced by a single global minima in the loss landscape. Further, we project the trajectories of the minimization onto the loss landscape for two different initial conditions. Irrespective of the initial values, the optimization converges rather quickly (in three to four steps) to the global minima. The optimized parameters are shown in Tab.3.

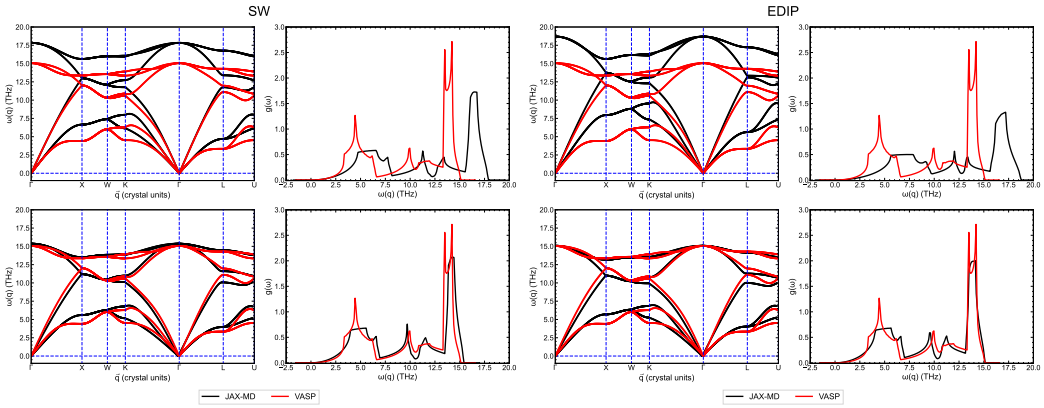

Figure 3: Phonon dispersion and VDOS before (top) and after (bottom) optimization for the SW and EDIP potentials.

The optimized force-field parameters are further evaluated by computing the phonon dispersion and vibrational density of states. It is worth noting that these properties can be directly computed from the force constants for which the force-field parameters are optimized. To this extent, the optimized Hessian is passed to the `phonopy` API in `python` for post-processing from which the phonon dispersion, vibrational density of states (VDOS), and thermodynamics properties are computed. Figure 3 shows the phonon dispersion and vibrational density of states (VDOS) before and after optimization for the SW Si and EDIP Si potential parameters. We observe that for both SW and EDIP potentials, the original parameters exhibit a poor match for VDOS and phonon dispersion. However, the optimized parameters exhibit a much closer match with the VDOS and phonon dispersion obtained from the DFT. It is worth noting that the potential parameters are optimized only for the force constants (Hessian) and not directly on the phonon dispersion or the VDOS. These results further reinforce the capability of automatic differentiation to optimize the force-field parameters toward any target property.

3.4 Radial distribution function

Till now, the properties we simulated were obtained based on energy minimization and represented the ground state properties at 0 K. However, most of the properties measurable by experiments are obtained through molecular dynamics simulation. To this extent, we now focus on optimizing the structure of the system at finite temperatures as represented by the radial distribution function (RDF). Note that the RDF captures both the short and medium-range structure of the material and is highly sensitive to the

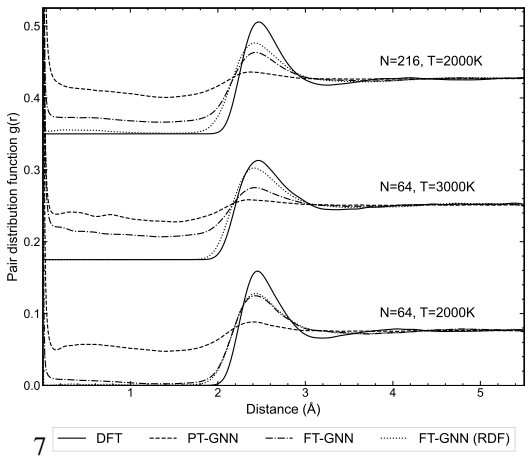

Figure 4: Comparison of the radial distribution function for the pre-trained GNN (PT-GNN), fine-tuned GNN (FT-GNN) and fine-tuned GNN with RDF (FT-GNN (RDF))

Table 5: Original and optimized parameters for SW and EDIP from multi-objective optimization.

Force field params	SW		EDIP	
	ϵ (eV)	λ	A (eV)	λ (eV)
Original	2.16826	21.0	7.9821730	1.4533108
CG	1.809324321148	34.619425564581	6.723988764513	1.396547822543

state of the material, that is, solid, liquid, or gas. Without loss of generality, we consider a message-passing graph neural network (MPGNN) pre-trained on the silicon energy and force dataset Cubuk et al. [2017] as described in A.2. We use this model as it is trained on finite temperature data and is expected to perform better than the classical potential. The simulation parameters and the model architecture are given above in section 2.5 and 2.3.

The RDF computed from AIMD with 64 silicon atoms at a temperature of 2000 K is used for fine-tuning. The MPGNN is fine-tuned by minimizing the error between the predicted RDF and the ground truth RDF (4). To test the transferability of this model, we evaluate the RDF at different temperatures, i.e., 1000 K and 3000 K, and a larger system of 216 Silicon atoms at 2000 K. For a more faithful evaluation, we also fine-tune MPGNN model only on the energy and forces at 2000 K. The comparison of the pre-trained (PT-GNN), GNN fine-tuned only on energy and forces (FT-GNN), and GNN fine-tuned on energy and forces along with RDF (FT-GNN (RDF)) is given in figure 4 with the quantitative error metrics presented in Tab. 4. We observe that the PT-GNN is unable to capture the behavior of the Si melt with high non-zero values even for low distances. While FT-GNN gives better performance than PT-GNN, we observe the value of RDF increases significantly closer to 0 distance. This suggests that the model is unable to capture the repulsions at short distances, a well-known problem with MLFFs [Bihani et al., 2024]. This is further exemplified by the poor performance of FT-GNN at higher temperatures of 3000 K and higher system sizes of 216 atoms. In contrast, we observe that the FT-GNN (RDF) perform significantly better at higher (unseen) temperatures and for larger systems. It is worth noting although the peaks at short distances are absent, the model still exhibits some non-zero values for the RDF at shorter distances. This suggests that the learned repulsive interaction by the FT-GNN (RDF) could be occasionally overcome by high velocity of atoms at high temperatures, a problem that could be addressed by including larger amounts of training data. More importantly, the results demonstrate that automatic differentiation could be used to fine-tune MLFFs or classical potentials towards structural properties obtained from dynamics simulations.

Table 4: Model evaluations for Training and Test data. The energy E and force F errors are RMSE values. The units for energy error are meV / Atom and for force are meV / Å / Atom.

	Training		Testing	
Models	E_{val}	F_{val}	E_{test}	F_{test}
Pre-trained	4.041	7.391	3.877	7.172
Fine-tuned	1.034	3.852	1.046	3.745
Fine-tuned (RDF)	1.070	3.834	1.078	3.689

3.5 Multi-objective optimization

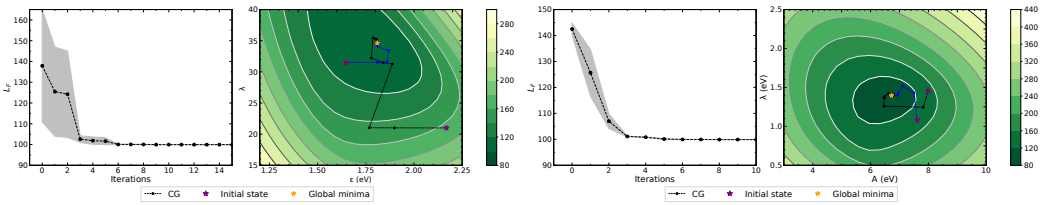

Figure 5: The learning curve and the optimization pathway in the parameter space with CG optimizer for SW (top) and EDIP (bottom) potentials for multi-objective optimization. The contour values are the values of the loss function at each parameter.

Table 6: Comparison of elastic constants C_{ij} for original and optimized parameters of SW and EDIP potential along with DFT values for multi-objective optimization.

Force field		C_{11} (GPa)	C_{12} (GPa)	C_{44} (GPa)
SW	Original	151.4131	76.4154	112.8927
	Optimized	153.4054	50.2369	133.8275
EDIP	Original	172.0389	64.6745	145.5683
	Optimized	153.4101	50.2358	133.822
DFT		153.2141	56.7969	149.6653

Finally, we evaluate the capability of automatic differentiation to optimize force fields on multiple properties. This is particularly useful when several experimental properties of a system are available, all of which could be computed through simulations. Here, we consider optimizing elastic tensor and force constants together. The combined loss function defined in (2), where Y_{ij} and $\hat{Y}_{ij}$ are matrix elements of C_{ij} while Z_{ij} and $\hat{Z}_{ij}$ are matrix elements of Hessian computed from force fields in JAX-MD and DFT respectively. The weights w_1 and w_2 give importance to the property, which is set to 1 in the present case.

Figure 5 shows the loss curve, loss landscape, and the optimization pathway for both SW and EDIP potentials. Similar to the earlier cases, we observe that the global minima are attained in very few iterations (three to six) of the CG minimizer, irrespective of the initialization. Further, we note that the optimized parameters for SW and EDIP potentials are distinctly different from those obtained from single objective optimization (see Tab. 5 for optimized parameters). This confirms the well-known observation that the parameters optimized from one property are not transferable to reproduce another property. This observation also emphasizes the relevance of multi-objective optimization using differentiable simulation for tuning the potential for target properties. The obtained values of elastic tensors C_{ij} are shown in Tab. 6. In the case of SW, we observe all the elastic tensors are closer to DFT values for the optimized parameters than the original one. In the case of EDIP, while C_{11} and C_{12} are closer, C_{44} is poorer for the optimized force field.

4 Discussion and conclusions

The extensive experiments presented thus far suggest that AD could be a powerful framework to tune force fields for atomistic simulations even for complex and non-linear properties such as vibrational density of states or radial distribution functions. Indeed, the optimization of pair-wise interactions have been explored in previous studies [Wang et al., 2023b] on LJ and Morse systems by fitting directly on RDFs. The framework presented here extends the approach to any property and systems including MLIPs. Beyond the tuning of force fields, we demonstrate that AD could be used to compute several properties accurately, which can be obtained directly as the gradients of simulations. This represents a paradigm shift from the classical finite difference and other numerical differentiation approaches widely used in atomistic simulations to compute properties.

The ability of AD to find global minima in a few iterations, albeit for a convex loss landscape, signifies the capability of the AD toward optimizing the force field parameters. Moreover, the simplicity of AD provides a strong test bed for evaluating the flexibility, generalizability, and robustness of force fields, both classical and machine-learned, to simulate complex materials, properties, and phases [Guan, 2022]. The fact that the parameters of the force field could be easily optimized to replicate any target property gives a strong handle to build a hypothesis and test what-if scenarios observed experimentally. Further, in the context of universal force fields [Merchant et al., 2023, Batatia et al., 2023], the approach could be used to either directly fine-tune the potentials to target properties or phases, or to simulate high temperature and pressure configurations or to simply to enhance the stability of the simulations by training on Hessian based properties or RDFs [Raja et al., 2024]. Moreover, the study demonstrates some of the "known" but interesting observations that the classical force fields, due to their limited parameter space, lack the flexibility to replicate multiple properties concurrently, emphasizing the relevance of MLFFs.

Indeed, the present work could supplement some of the several earlier efforts to build coarse-grained models [King et al., 2024a, Wang et al., 2023b, Majewski et al., 2023, Wang, 2022] and fine-tune potentials on experimental datasets [Thaler and Zavadlav, 2021]. AD could be seamlessly used to compute properties in the forward simulation and optimize them against ground truth by computing the automatic gradients. These ground truths could be directly obtained from first-principle simulations, experiments, or even manually constructed by averaging some features. For instance, a major challenge in atomistic simulations is the development of coarse-grained models. To this extent, AD could be used to construct such coarse-grained models by optimizing on a "synthetic data" generated to capture the averaged properties from full-atom simulations [Wang et al., 2020,

2023b]. Some examples could be the potential for united atom models against the all-atom models or even the bead-string models developed from all-atom models. AD provides a principled way towards attacking these problems ensuring the guarantee of reaching the solution provided it exists.

In conclusion, our study highlights the potential of end-to-end differentiable atomistic simulations for both rapid and precise force field optimization through analytical gradient computation of simulations. The demonstrated applications not only enhance the efficiency of force field refinement but also extend toward computing critical properties, such as derivatives of ensemble averages [Zu and Goodrich, 2024]. By bridging the gap between machine learning and physics, differentiable simulations provides a fillip to optimization and control in atomistic simulations, setting the stage for broader applications in several areas including insilico materials discovery, self-assembly, topological and material optimization, and enhanced sampling. It is worth noting that there are several limitations for the present work. The work is carried out on simple unary Si system. Extending this to multicomponent systems with complex behavior requires further exploration. Similarly, extending differentiable simulations to more complex properties such as diffusion constant, thermal expansions, etc. would be part of future exploration.

Acknowledgments and Disclosure of Funding

M.B. acknowledges the National Science Foundation under Grant No. DMREF- 1922167.

References

Simon Axelrod, Daniel Schwalbe-Koda, Somesh Mohapatra, James Damewood, Kevin P Greenman, and Rafael Gómez-Bombarelli. Learning matter: Materials design with machine learning and atomistic simulations. *Accounts of Materials Research*, 3(3):343–357, 2022.

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

Martin Z. Bazant, Efthimios Kaxiras, and J. F. Justo. Environment-dependent interatomic potential for bulk silicon. *Phys. Rev. B*, 56:8542–8552, Oct 1997. doi: 10.1103/PhysRevB.56.8542. URL `https://link.aps.org/doi/10.1103/PhysRevB.56.8542`.

Vaibhav Bihani, Sajid Mannan, Utkarsh Pratiush, Tao Du, Zhimin Chen, Santiago Miret, Matthieu Micoulaut, Morten M Smedskjaer, Sayan Ranu, and NM Anoop Krishnan. Egraffbench: evaluation of equivariant graph neural network force fields for atomistic simulations. *Digital Discovery*, 3(4): 759–768, 2024.

Suresh Bishnoi, Ravinder Bhattoo, Sayan Ranu, and NM Anoop Krishnan. Discovering symbolic laws directly from trajectories with hamiltonian graph neural networks. *Machine Learning: Science and Technology*, 2023.

Suresh Bishnoi, Jayadeva Jayadeva, Sayan Ranu, and NM Anoop Krishnan. Brognet: Momentum-conserving graph neural stochastic differential equation for learning brownian dynamics. In *The Twelfth International Conference on Learning Representations*, 2024.

P. E. Blöchl. Projector augmented-wave method. *Phys. Rev. B*, 50:17953–17979, Dec 1994. doi: 10.1103/PhysRevB.50.17953. URL `https://link.aps.org/doi/10.1103/PhysRevB.50.17953`.

Michael P Brenner and Ella M King. Scientific uses of automatic differentiation. *Collections*, 56(04), 2023.

Manuel Carrer, Henrique Musseli Cezar, Sigbjørn Løland Bore, Morten Ledum, and Michele Cascella. Learning force field parameters from differentiable particle-field molecular dynamics. *Journal of Chemical Information and Modeling*, 64(14):5510–5520, 2024. doi: 10.1021/acs.jcim.4c00564. URL `https://doi.org/10.1021/acs.jcim.4c00564`. PMID: 38963184.

Siwei Chen, Yiqing Xu, Cunjun Yu, Linfeng Li, Xiao Ma, Zhongwen Xu, and David Hsu. Daxbench: Benchmarking deformable object manipulation with differentiable physics. *arXiv preprint arXiv:2210.13066*, 2022.

Ekin D. Cubuk, Brad D. Malone, Berk Onat, Amos Waterland, and Efthimios Kaxiras. Representations in neural network based empirical potentials. *The Journal of Chemical Physics*, 147(2): 024104, 07 2017. ISSN 0021-9606. doi: 10.1063/1.4990503. URL https://doi.org/10.1063/1.4990503.

Volker L Deringer, Miguel A Caro, and Gábor Csányi. Machine learning interatomic potentials as emerging tools for materials science. *Advanced Materials*, 31(46):1902765, 2019.

Axel Dürrbaum, Willy Klier, and Hubert Hahn. Comparison of automatic and symbolic differentiation in mathematical modeling and computer simulation of rigid-body systems. *Multibody System Dynamics*, 7:331–355, 2002.

Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

Pin-Wen Guan. Differentiable thermodynamic modeling. *Scripta Materialia*, 207:114217, 2022. ISSN 1359-6462. doi: https://doi.org/10.1016/j.scriptamat.2021.114217. URL https://www.sciencedirect.com/science/article/pii/S1359646221004978.

William G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31: 1695–1697, Mar 1985. doi: 10.1103/PhysRevA.31.1695. URL https://link.aps.org/doi/10.1103/PhysRevA.31.1695.

Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*, 2019.

João F. Justo, Martin Z. Bazant, Efthimios Kaxiras, V. V. Bulatov, and Sidney Yip. Interatomic potential for silicon defects and disordered phases. *Phys. Rev. B*, 58:2539–2550, Aug 1998. doi: 10.1103/PhysRevB.58.2539. URL https://link.aps.org/doi/10.1103/PhysRevB.58.2539.

Ella M King, Chrisy Xiyu Du, Qian-Ze Zhu, Samuel S Schoenholz, and Michael P Brenner. Programming patchy particles for materials assembly design. *Proceedings of the National Academy of Sciences*, 121(27):e2311891121, 2024a.

Ella M. King, Megan C. Engel, Caroline Martin, Alp M. Sunol, Qian-Ze Zhu, Sam S. Schoenholz, Vinothan N. Manoharan, and Michael P. Brenner. Inferring interaction potentials from stochastic particle trajectories, 2024b. URL https://arxiv.org/abs/2406.01522.

G. Kresse and J. Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, 54:11169–11186, Oct 1996. doi: 10.1103/PhysRevB.54.11169. URL https://link.aps.org/doi/10.1103/PhysRevB.54.11169.

G. Kresse and J. Hafner. Ab initio molecular dynamics for liquid metals. *Phys. Rev. B*, 47:558–561, Jan 1993. doi: 10.1103/PhysRevB.47.558. URL https://link.aps.org/doi/10.1103/PhysRevB.47.558.

G. Kresse and D. Joubert. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B*, 59:1758–1775, Jan 1999. doi: 10.1103/PhysRevB.59.1758. URL https://link.aps.org/doi/10.1103/PhysRevB.59.1758.

Maciej Majewski, Adrià Pérez, Philipp Thölke, Stefan Doerr, Nicholas E. Charron, Toni Giorgino, Brooke E. Husic, Cecilia Clementi, Frank Noé, and Gianni De Fabritiis. Machine learning coarse-grained potentials of protein thermodynamics. *Nature Communications*, 14(1):5739, Sep 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-41343-1. URL https://doi.org/10.1038/s41467-023-41343-1.

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017.

J.-D. Müller and P. Cusdin. On the performance of discrete adjoint cfd codes using automatic differentiation. *International Journal for Numerical Methods in Fluids*, 47(8-9):939–945, 2005. doi: https://doi.org/10.1002/fld.885. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.885`.

Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual review of physical chemistry*, 71(1):361–390, 2020.

Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 1984. doi: 10.1063/1.447334. URL `https://doi.org/10.1063/1.447334`.

Alfonso Pedone, Marco Bertani, Luca Brugnoli, and Annalisa Pallini. Interatomic potentials for oxide glasses: Past, present, and future. *Journal of Non-Crystalline Solids: X*, 15:100115, 2022.

John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996. doi: 10.1103/PhysRevLett.77.3865. URL `https://link.aps.org/doi/10.1103/PhysRevLett.77.3865`.

GP Purja Pun and Y Mishin. Development of an interatomic potential for the ni-al system. *Philosophical Magazine*, 89(34-36):3245–3267, 2009.

Sanjeev Raja, Ishan Amin, Fabian Pedregosa, and Aditi S Krishnapriyan. Stability-aware training of neural network interatomic potentials with differentiable boltzmann estimators. *arXiv preprint arXiv:2402.13984*, 2024.

Sebastien Röcken and Julija Zavadlav. Accurate machine learning force fields via experimental and simulation data fusion. *npj Computational Materials*, 10(1):69, Apr 2024. ISSN 2057-3960. doi: 10.1038/s41524-024-01251-4. URL `https://doi.org/10.1038/s41524-024-01251-4`.

Srikanth Sastry and C Austen Angell. Liquid–liquid phase transition in supercooled silicon. *Nature materials*, 2(11):739–743, 2003.

Samuel S. Schoenholz and Ekin D. Cubuk. Jax m.d. a framework for differentiable physics. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL `https://papers.nips.cc/paper/2020/file/83d3d4b6c9579515e1679aca8cbc8033-Paper.pdf`.

Daniel Schwalbe-Koda, Aik Rui Tan, and Rafael Gómez-Bombarelli. Differentiable sampling of molecular geometries with uncertainty-based adversarial attacks. *Nature communications*, 12(1):5104, 2021.

Frank H. Stillinger and Thomas A. Weber. Computer simulation of local order in condensed phases of silicon. *Phys. Rev. B*, 31:5262–5271, Apr 1985. doi: 10.1103/PhysRevB.31.5262. URL `https://link.aps.org/doi/10.1103/PhysRevB.31.5262`.

John C Strikwerda. *Finite difference schemes and partial differential equations*. SIAM, 2004.

Teresa Tamayo-Mendoza, Christoph Kreisbeck, Roland Lindh, and AlÃ¡n Aspuru-Guzik. Automatic differentiation in quantum chemistry with applications to fully variational hartreeâ€“fock. *ACS Central Science*, 4(5):559–566, 2018. doi: 10.1021/acscentsci.7b00586. URL `https://doi.org/10.1021/acscentsci.7b00586`. PMID: 29806002.

Stephan Thaler and Julija Zavadlav. Learning neural network potentials from experimental data via differentiable trajectory reweighting. *Nature communications*, 12(1):6884, 2021.

Abishek Thangamuthu, Gunjan Kumar, Suresh Bishnoi, Ravinder Bhattoo, NM Krishnan, and Sayan Ranu. Unravelling the performance of physics-informed graph neural networks for dynamical systems. *Advances in Neural Information Processing Systems*, 35:3691–3702, 2022.

Jeffrey P. Thomas, Earl H. Dowell, and Kenneth C. Hall. Using automatic differentiation to create a nonlinear reduced-order-model aerodynamic solver. *AIAA Journal*, 48(1):19–24, 2010. doi: 10.2514/1.36414. URL https://doi.org/10.2514/1.36414.

A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.*, 271:108171, 2022. doi: 10.1016/j.cpc.2021.108171.

Moritz Thürlemann and Sereina Riniker. Hybrid classical/machine-learning force fields for the accurate description of molecular condensed-phase systems. *Chem. Sci.*, 14:12661–12675, 2023. doi: 10.1039/D3SC04317G. URL http://dx.doi.org/10.1039/D3SC04317G.

A Togo and I Tanaka. First principles phonon calculations in materials science. *Scr. Mater.*, 108:1–5, Nov 2015.

Iam Torrens. *Interatomic potentials*. Elsevier, 2012.

Mengyi Wang, NM Anoop Krishnan, Bu Wang, Morten M Smedskjaer, John C Mauro, and Mathieu Bauchy. A new transferable interatomic potential for molecular dynamics simulations of borosilicate glasses. *Journal of Non-Crystalline Solids*, 498:294–304, 2018.

Wujie Wang. *Differentiable Multiscale Molecular Simulations*. PhD thesis, Massachusetts Institute of Technology. Department of Materials Science and Engineering, 2022. URL https://dspace.mit.edu/handle/1721.1/147220.

Wujie Wang, Simon Axelrod, and Rafael Gómez-Bombarelli. Differentiable molecular simulations for control and learning. *arXiv preprint arXiv:2003.00868*, 2020.

Wujie Wang, Zhenghao Wu, Johannes C. B. Dietschreit, and Rafael GÃ³mez-Bombarelli. Learning pair potentials using differentiable simulations. *The Journal of Chemical Physics*, 158(4):044113, 01 2023a. ISSN 0021-9606. doi: 10.1063/5.0126475. URL https://doi.org/10.1063/5.0126475.

Wujie Wang, Zhenghao Wu, Johannes CB Dietschreit, and Rafael Gómez-Bombarelli. Learning pair potentials using differentiable simulations. *The Journal of Chemical Physics*, 158(4), 2023b.

Fabian Zills, Moritz René Schäfer, Nico Segreto, Johannes Kästner, Christian Holm, and Samuel Tovey. Collaboration on machine-learned potentials with ipsuite: A modular framework for learning-on-the-fly. *The Journal of Physical Chemistry B*, 128(15):3662–3676, 2024. doi: 10.1021/acs.jpcb.3c07187. URL https://doi.org/10.1021/acs.jpcb.3c07187. PMID: 38568231.

Mengjie Zu and Carl P. Goodrich. Designing athermal disordered solids with automatic differentiation. *Communications Materials*, 5(1):141, Aug 2024. ISSN 2662-4443. doi: 10.1038/s43246-024-00583-4. URL https://doi.org/10.1038/s43246-024-00583-4.

A Appendix / supplemental material

A.1 VASP settings

The standard VASP pseudo-potential (PSP) based on the projector augmented wave (PAW) [Blöchl, 1994, Kresse and Joubert, 1999] is used for Si. For the exchange-correlation functional, the Perdew-Burke-Ernzerhof (PBE) with generalized gradient approximation (GGA) [Perdew et al., 1996] is employed. The Si PSP has the 4 valence electrons with the $3s^2 3p^2$ as the configuration. All the simulations are carried out using periodic boundary conditions (PBC).

The phonon and elastic constants are computed at 0 K through molecular statics employing energy minimization. For computing phonon and elastic constants, a plane-wave cutoff of 520 eV is used with tighter energy and force convergence criterion of 10^{-9} eV and 10^{-7} eV/Å, respectively. This along with a Γ-centered *k*-point mesh of $12 \times 12 \times 12$ ensures accurate convergence of these properties. Gaussian smearing of 0.05 eV is employed and spin-polarization is not considered as the Si phases simulated have a non-magnetic ground state. Elastic constants are computed on a unit cell with 8 Si atoms. For phonon calculations, a super-cell that is $2 \times 2 \times 2$ the unit-cell yields converged phonon frequencies. The *k*-point mesh is adjusted for the super-cell to keep the same *k*-point density. Generation of displacement and post-processing of the force constant data is performed with `phonopy` [Togo and Tanaka, 2015].

For evaluating the performance of the potential in molecular dynamics simulations, finite temperature *ab initio* molecular dynamics (AIMD) is employed to generate the ground truth. For AIMD simulations, a lower cutoff of 400 eV is used along with an energy convergence criterion of 10^{-6} eV. A super-cell that is $2 \times 2 \times 2$ the unit-cell is used as a starting point. $2 \times 2 \times 2$ Γ-centered *k*-point mesh is used to obtain accurate forces as Silicon in the liquid phase exhibits a metallic nature, that is, the density of states at Fermi energy is non-zero. A time step of 1 fs is used for integration of the equation of motion. The temperature is controlled using the Nosé–Hoover thermostat [Nosé, 1984, Hoover, 1985]. The crystal is equilibrated at 2000 K and the initial 5000 steps (5 ps) for melting are discarded to avoid any remnants of the crystal structure. The radial distribution function (RDF) obtained from the simulation is used as the ground truth representing the structure of the Si system. The averaging is done based on the steps used in the differentiable simulation setup. The process is repeated for additional temperatures of 1000 K and 3000 K and a larger cell (that is, $3 \times 3 \times 3$ the unit-cell). These structures are maintained as the test set to evaluate the generalizability of the force field to unseen temperatures.

A.2 Model details

Stillinger-Weber potential

The Stillinger-Weber (SW) potential is used to describe the atomistic behavior of bulk Silicon (Si) Stillinger and Weber [1985]. The potential energy of all the atoms in a bulk Si system is given by:

$$U_{Total} = \sum_i \sum_{j>i} \phi_2(r_{ij}) + \sum_i \sum_{j \neq i} \sum_{k>j} \phi_3(\vec{r}_{ij}, \vec{r}_{ik}, \theta_{ijk}) \tag{9}$$

where the ϕ_2 and ϕ_3 are the two and three-body interaction terms that have the following functional form:

$$\phi_2(r_{ij}) = A_{ij}\epsilon_{ij}\left[B_{ij}\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{q_{ij}}\right]\exp\left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right) \tag{10}$$

the first term in 10 repulsive part while the second term is the attractive part.

$$\phi_3(\vec{r}_{ij}, \vec{r}_{ik}, \theta_{ijk}) = \lambda_{ijk}\epsilon_{ijk}[\cos\theta_{ijk} - \cos\theta_{0ijk}]^2 \exp\left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right)\exp\left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}}\right) \tag{11}$$

The parameters ϵ and σ determine the energy and distance scaling respectively. The A and B parameter is chosen so that the minimum of the $\phi_2(r_{ij})$ i.e $r_0 = 2^{1/6}\sigma$ has the energy $-\epsilon$. Here r_0 is the equilibrium bond length between two atoms and $r_c = a\sigma$ is the cutoff for the potential beyond which the potential energy is zero. θ_{0ijk} is the equilibrium bond angle between triplets of atoms. The λ controls the strength of the three-body term and .γ controls the rate at which the three-body term

Table 7: Original parameters of the SW potential for Si.

Params	Values	Params	Values
ϵ (eV)	2.16826	A	7.049556277
σ (Å)	2.0951	B	0.6022245584
a	1.80	p	4.0
λ	21.0	q	0.0
γ	1.20		

goes to zero. The p and q are free parameters that control the curvature of the repulsive part and the attractive part respectively. The $\theta_{0ijk} = 109.47°$ is the tetrahedral bond angle which ensures that the diamond structure is in the minimum energy (ground) state.

Environment dependent interatomic potential

Environment-dependent interatomic potential (EDIP) is a bond order potential originally used for bulk silicon Bazant et al. [1997], Justo et al. [1998]. The pairwise potential energy U_i depends on the local environment of the atom. This is taken into account by bond order functions which depend on the local coordination number. This way the potential is able to capture the different types of chemical bonding for different coordinations for different phases of Silicon.

The potential energy is the sum of two-body and three-body terms and is given by -

$$U_i = \sum_{j \neq i} \phi_2(r_{ij}) + \sum_{j \neq i} \sum_{k \neq i, k > j} \phi_3(\vec{r}_{ij}, \vec{r}_{ik}, \theta_{ijk}) \tag{12}$$

where Z_i the effective co-ordination number is defined by -

$$Z_i = \sum_{m \neq i} f(r_{im}) \tag{13}$$

The sum is carried over the neighbors of the atom. Here $f(R_{im})$ is the cutoff function which drops smoothly to 0 from 1 from a to c. Here a and c are outer and inner cutoffs. This function depends only on the separation between the atoms r.

$$f(r) = \begin{cases} 1 & r \leq c \\ \exp(\frac{\alpha}{1-x^{-3}}) & c < r < a \\ 0 & a \geq r \end{cases} \tag{14}$$

where the x takes the form -

$$x = \frac{r - c}{a - c} \tag{15}$$

The two-body potential is given by -

$$\phi_2(r, Z) = A \left[(\frac{B}{r})^{\rho} - p(Z) \right] \exp\left(\frac{\sigma}{r - a} \right) \tag{16}$$

Here $p(Z)$ is the Gaussian function which depends on the effective local coordination Z.

$$p(Z) = \exp(-\beta Z^2) \tag{17}$$

The three-body term is given by -

$$\phi_3(\vec{r}_{ij}, \vec{r}_{ik}, Z_i) = g(r_{ij}) g(r_{ik}) h(l_{ijk}, Z_i) \tag{18}$$

where $l_{ijk} = \cos(\theta_{ijk}) = \vec{r}_{ij} \cdot \vec{r}_{ik} / r_{ij} r_{ik}$

Table 8: Original parameters of the EDIP potential for Si.

Params	Values	Params	Values
u_1	-0.165799	α	3.1083847
u_2	32.557	A	7.9821730
u_3	0.286198	λ	1.4533108
u_4	0.66	B	1.5075463
ρ	1.2085196	γ	1.1247945
η	0.2523244	σ	0.5774108
Q_0	312.1341346	c	2.5609104
μ	0.6966326	a	3.1213820
β	0.0070975		

The radial part $g(r)$ is given by -

$$g(r) = \exp\left(\frac{\gamma}{r-a}\right) \tag{19}$$

The angular part of the function $h(l, Z)$ depends on the effective local coordination and is described as -

$$h(l, Z) = \lambda[(1 - \exp(-Q(Z)(l + \tau(Z))^2)) + \eta Q(Z)(l + \tau(Z))^2] \tag{20}$$

where $Q(Z) = Q_0 \exp(-\mu Z)$ and the function $\tau(Z)$ is given by -

$$\tau(Z) = u_1 + u_2(u_3 \exp(-u_4 Z) - \exp(-2u_4 Z)) \tag{21}$$

A and λ determine the energy scales of the two-body and three-body terms respectively. σ determines the distance scale between neighbors while B and γ are parameters of the repulsive two-body term and radial three-body term respectively. ρ is the exponent for the repulsive part of the two-body term while β is a parameter for the two-body bond order function $p(Z)$. The two functions also depend on Z. u_1, u_2, u_3 and u_4 are parameter for the three-body bond order function $\tau(Z)$. Q_0 and μ are parameters for the three-body bond order function Q(Z) while η is a parameter for $h(l, Z)$. a and c are the outer and inner cutoff for the function $f(r)$ and α the exponential scaling parameter.

Machine learned interatomic potential

Consider a system with N particles. The system is represented as a graph $G = (V, E)$ where:

- $V = \{v_1, v_2, \ldots, v_N\}$ is the set of nodes representing particles.
- $E \subseteq V \times V$ is the set of edges representing particle interactions.

Each node v_i has a feature vector $\mathbf{h}_i$, and each edge (v_i, v_j) has a feature vector $\mathbf{e}_{ij}$.

The GNN consists of multiple layers, each performing message passing and updating node features.

$$\mathbf{m}_i^{(k)} = \sum_{j \in \mathcal{N}(i)} F_{\text{msg}}^{(k)}(\mathbf{h}_i^{(k)}, \mathbf{h}_j^{(k)}, \mathbf{e}_{ij}) \tag{22}$$

Here, $\mathbf{m}_i^{(k)}$ is the message for node i at layer k, $\mathbf{h}_i^{(k)}$ and $\mathbf{h}_j^{(k)}$ are the node features, and $\mathbf{e}_{ij}$ is the edge feature.

$$\mathbf{h}_i^{(k+1)} = F_{\text{update}}^{(k)}(\mathbf{h}_i^{(k)}, \mathbf{m}_i^{(k)}) \tag{23}$$

Here, $\mathbf{h}_i^{(k+1)}$ is the updated feature for node i at layer $k+1$.

The global state is updated based on the final node features:

$$\mathbf{g} = \text{Aggregate}(\{\mathbf{h}_i^{(K)}\}) \tag{24}$$

Here, $\mathbf{g}$ is the global state obtained by aggregating the final node features $\mathbf{h}_i^{(K)}$.

The energy U is computed using a decoder applied to the global state:

$$U = D(\mathbf{g}) \tag{25}$$

Here, D is the function that maps the global state to the system's potential energy and K is the total number of message-passing steps. The functions F_{update}, F_{msg}, and D are neural networks and their weights are learned during the training.

A.3 Property computation

The simplest way to compute the vibrational properties of a crystal is to use the harmonic approximation whereby we assume that the atoms vibrate around their equilibrium positions. This approximation works when the displacements are small (~ 0.01 Å) at constant volume. We can perform a taylor expansion of the potential energy of the crystal as a function of displacements $u_\alpha(lp)$:

$$U = \Phi_0 + \sum_{lp}\sum_{\alpha}\Phi_\alpha(lp)u_\alpha(lp) + \frac{1}{2}\sum_{lpl'p'}\sum_{\alpha\beta}\Phi_{\alpha\beta}(lp,l'p')u_\alpha(lp)u_\beta(l'p') + O(u^3) \tag{26}$$

Here the labels l and p correspond to the unit cell and the atoms respectively. The $\alpha, \beta, ..$ correspond to the Cartesian indices, and the $\Phi_{1,2,..,n}$ are the force constants of nth order. We can drop the $O(u^3)$ terms as the displacements are small. At the equilibrium positions $R_\alpha(lp)$, the forces $F_\alpha(lp)$ are zero, and thus:

$$\Phi_\alpha(lp) = -F_\alpha(lp) = \frac{\partial U}{\partial R_\alpha(lp)} = 0 \tag{27}$$

and the second order force constants $\Phi_{\alpha\beta}(lp, l'p')$ are given as:

$$\Phi_{\alpha\beta}(lp,l'p') = -\frac{\partial F_\beta(l'p')}{\partial R_\alpha(lp)} = \frac{\partial^2 U}{\partial R_\alpha(lp)\partial R_\beta(l'p')} \tag{28}$$

Since we do not know the analytic force constants they are calculated by applying a finite displacement (FD) to atoms and calculating the resulting forces on other atoms as:

$$\Phi_{\alpha\beta}(lp,l'p') \simeq -\frac{F_\beta(l'p', \Delta R_\alpha(lp)) - F_\beta(l'p')}{\Delta R_\alpha(lp)} \tag{29}$$

In theory, the force constant matrix (hessian) mentioned above should be computed for an infinite lattice however in practice we use another approximation that the induced forces due to displacements of atoms far away are small or close to zero (Frozen phonon approximation). For N atoms this leaves us with a total of $3N$ force calculations or $6N$ if the centering difference method is used. For materials where the force constants decay slowly as a function of the super-cell size, it becomes computationally challenging to perform a large number of force calculations, especially when using *ab initio* methods like DFT. Hence, modern packages like `phonopy` Togo and Tanaka [2015] use symmetry to reduce the number of force calculations and also improve numerical stability.

Once the force constant matrix is calculated we can set up the dynamical matrix in the reciprocal space:

$$D^{\alpha\beta}_{pp'}(q) = \sum_{l'}\frac{\Phi_{\alpha\beta}(0p,l'p')}{\sqrt{m_p m_{p'}}}\exp(iq\cdot[R(l'p') - R(0p)]) \tag{30}$$

where q is the phonon wave vector and m_p is the mass of the atom p. With the approximations made we can set up an eigenvalue problem to calculate the phonon frequencies ω_{qj} and the phonon eigenvectors e_{qj} as follows:

$$\sum_{\beta p'} D^{\alpha\beta}_{pp'}(q) e^{\beta p'}_{qj} = \omega^2_{qj} e^{\alpha p}_{qj} \tag{31}$$

Here j is the band index. For a 3-dimensional system, the dynamical matrix is $3N \times 3N$ hermitian. The phonon density of states (PDOS) then can be calculated as:

$$g(\omega) = \frac{1}{l_{max}} \sum_{qj} \delta(\omega - \omega_{qj}) \tag{32}$$

The l_{max} is the number of unit cells used. This ensures that $\int g(\omega) d\omega = 3N$. Using the obtained phonon frequencies we can compute the phonon number n and the energy of the phonon system E_{ph} under the harmonic approximation:

$$n = \frac{1}{\exp(\frac{\hbar\omega_{qj}}{k_B T}) - 1} \tag{33}$$

$$E_{ph} = \sum_{qj} \hbar\omega_{qj} \left[\frac{1}{2} + \frac{1}{\exp(\frac{\hbar\omega_{qj}}{k_B T}) - 1} \right] \tag{34}$$

Where $\hbar$, k_B and T are the Plank's constant, Boltzmann constant, and the temperature respectively. Using the known relations in statistical mechanics we can further compute the heat capacity at constant volume (C_v):

$$C_v = \left(\frac{\partial E}{\partial T} \right)_V = \sum_{qj} k_B \left(\frac{\hbar\omega_{qj}}{k_B T} \right)^2 \frac{\exp(\frac{\hbar\omega_{qj}}{k_B T})}{[\exp(\frac{\hbar\omega_{qj}}{k_B T}) - 1]^2} \tag{35}$$

Using the canonical partition function Z the free energy F and entropy S can be computed:

$$Z = \exp(\frac{-\psi}{k_B T}) \prod_{qj} \frac{\exp(\frac{-\hbar\omega_{qj}}{2k_B T})}{1 - \exp(\frac{-\hbar\omega_{qj}}{k_B T})} \tag{36}$$

$$F = -k_B T \ln Z = \psi + \frac{1}{2} \sum_{qj} \hbar\omega_{qj} + k_B T \sum_{qj} \ln[1 - \exp(\frac{-\hbar\omega_{qj}}{k_B T})] \tag{37}$$

$$S = -\frac{\partial F}{\partial T} = \frac{1}{2T} \sum_{qj} \hbar\omega_{qj} \coth(\frac{\hbar\omega_{qj}}{2k_B T}) - k_B \sum_{qj} \ln[2 \sinh(\frac{\hbar\omega_{qj}}{2k_B T})] \tag{38}$$

In JAX-MD we have access to analytical gradients using AD and thus we do not have to perform any approximation in computing the force constants like in eq 29. This can be of particular advantage as the choice of displacement can alter the results. A large displacement will mean that anharmonicities are introduced while a small displacement will underestimate the curvature. AD force constants are also more numerically stable. Post-processing of both force constants is done by `phonopy` [Togo and Tanaka, 2015] as mentioned before.

The details for the calculation and the implementation of elastic constants and RDF can be found in the `elasticity.py` module and `quantity.pair_correlation` function in JAX-MD respectively.

A.4 Code snippet

Code for optimizing the force field for target elastic properties.

```
import jax.numpy as jnp
from jax import grad, jit
from jax_md import space, energy, quantity, elasticity

# Coordinates
R = jnp.array([[0.25, 0.75, 0.25],
       [0.  , 0.  , 0.5 ],
       [0.25, 0.25, 0.75],
       [0.  , 0.5 , 0.  ],
       [0.75, 0.75, 0.75],
       [0.5 , 0.  , 0.  ],
       [0.75, 0.25, 0.25],
       [0.5 , 0.5 , 0.5 ]])

# Lattice vector
latvec = jnp.array([[5.43000, 0.00000, 0.00000],
                    [0.00000, 5.43000, 0.00000],
                    [0.00000, 0.00000, 5.43000]])

# Parameters of forcefield
params = jnp.array([2.16826, 21.0])

# Reference elastic tensor
C_t = jnp.array([[153.21414,56.79690,56.79690,0,0,0],
                 [56.79690,153.21414,56.79690,0,0,0],
                 [56.79690,56.79690,153.21414,0,0,0],
                 [0,0,0,149.66528,0,0],
                 [0,0,0,0,149.66528,0],
                 [0,0,0,0,0,149.66528]])

def setup(R, latvec, C_t):
  # Setup simulation environment
  displacement, shift = space.periodic_general(latvec)
  dist_fun = space.metric(displacement)

  @jit
  def run(params):
    # Function to calculate the elastic tensor

    # Define the energy function
    energy_fn = energy.stillinger_weber(displacement,epsilon=params
   [0], lam=params[1])

    # Solve for elastic tensor
    emt_fn = elasticity.athermal_moduli(energy_fn)
    C = emt_fn(R, latvec)
    C_p = jit(elasticity.tensor_to_mandel)(C)

    # Return the error between predicted and true elastic tensor
    # based on Frobenius Norm
    return jnp.linalg.norm(C_p - C_t, ord='fro')
  return run

run = setup(R, latvec, C_t)

# Computes the elastic tensor for given params
run(params)

# Computes the gradient of elastic tensor w.r.t to params
grad(run)(params)
```