

Stress and heat flux via automatic differentiation

Marcel F. Langer,^{1,2,3} J. Thorben Frank,^{1,2} and Florian Knoop⁴

¹*Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany*

²*BIFOLD – Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*

³*The NOMAD Laboratory at the Fritz Haber Institute of the Max Planck Society and Humboldt University, Berlin, Germany*

⁴*Theoretical Physics Division, Department of Physics, Chemistry and Biology (IFM), Linköping University, SE-581 83 Linköping, Sweden*

(*Electronic mail: mail@marcel.science)

(Dated: 3 May 2023)

Machine-learning potentials provide computationally efficient and accurate approximations of the Born-Oppenheimer potential energy surface. This potential determines many materials properties and simulation techniques usually require its gradients, in particular forces and stress for molecular dynamics, and heat flux for thermal transport properties. Recently developed potentials feature high body order and can include equivariant semi-local interactions through message-passing mechanisms. Due to their complex functional forms, they rely on automatic differentiation (AD), overcoming the need for manual implementations or finite-difference schemes to evaluate gradients. This study demonstrates a unified AD approach to obtain forces, stress, and heat flux for such potentials, and provides a model-independent implementation. The method is tested on the Lennard-Jones potential, and then applied to predict cohesive properties and thermal conductivity of tin selenide using an equivariant message-passing neural network potential.

I. INTRODUCTION

Molecular dynamics (MD) simulations enable computational prediction of thermodynamic quantities for a wide range of quantum systems, and constitute a cornerstone of modern computational science¹. In MD, systems are simulated by propagating Newton’s equations of motion – potentially modified to model statistical ensembles – numerically in time, based on the forces acting on each atom due to their movement on the Born-Oppenheimer potential-energy surface (PES). Therefore, the quality of the underlying PES is important for the predictive ability of this method. First-principles electronic structure methods such as density-functional theory (DFT) can be used to perform high-accuracy MD simulations², provided the exchange-correlation approximation is reliable³. Such approaches are, however, restricted by high computational cost, severely limiting accessible size and time scales. Computationally efficient approximations to the underlying PES are therefore required for the atomistic simulation of larger systems: Forcefields (FFs) are built on analytical functional forms that are often based on physical bonding principles, parametrized to reproduce quantities of interest for a given material⁴. They are computationally cheap, but parametrizations for novel materials are not always available, and their flexibility is limited by their fixed functional form. Machine-learning potentials (MLPs)^{5–11}, where a potential is inferred based on a small set of reference calculations, aim to retain near first-principles accuracy while remaining linear scaling with the number of atoms. While MLPs are limited, in principle, to modeling the physical mechanisms present in the training data, they have nevertheless emerged as an important tool for MD^{12–16}, often combined with active learning schemes^{17–21}. Modern MLPs can include semi-local interactions through message-passing (MP) mechanisms²², internal equivariant representations²³, and body-order expansions²⁴, which enable the efficient construction of flexible many-body

interactions. In such complex architectures, the manual implementation of derivatives is often unfeasible. Finite-difference approaches require tuning of additional parameters, as well as repeated energy evaluations. Automatic differentiation (AD)^{25,26} presents an intriguing alternative: If the computation of the potential energy U is implemented in a suitable framework, derivatives such as the forces \mathbf{F} or the stress $\boldsymbol{\sigma}$ can be computed with the same asymptotic computational cost as computing the energy. This is accomplished by decomposing the underlying ‘forward’ computation into elementary operations with analytically tractable derivatives, computing local gradients for a given input, and then combining the resulting values using the chain rule.

This work provides a systematic discussion of the use of AD to compute forces, stress, and heat flux for MLPs. While the calculation of forces with AD is common practice^{27–30}, stress and heat flux are not yet commonly available for many MLPs. Both quantities have been the focus of much previous work due to the difficulty of defining and implementing them for many-body potentials and periodic boundary conditions^{31–42}. Introducing an abstract definition of MLPs as functions of a graph representation of atomistic systems, unified formulations of stress and heat flux are given, which can be implemented generally for any such graph-based machine-learning potential (GLP). An example implementation using `jax`⁴³ is provided in the `g1p` package⁴⁴. To validate the approach, different formulations of stress and heat flux are compared for the Lennard-Jones potential⁴⁵, where analytical derivatives are readily available for comparison, as well as a state-of-the-art message-passing neural network (MPNN), `SO3KRATES`⁴⁶. Having established the correctness of the proposed implementation, the ability of `SO3KRATES` to reproduce first-principles cohesive properties and thermal conductivity of tin selenide (SnSe) is studied.

II. AUTOMATIC DIFFERENTIATION

Automatic differentiation (AD) is a technique to obtain derivatives of functions implemented as computer programs.^{25,26} It is distinct from *numerical differentiation*, where finite-difference schemes are employed, and *symbolic differentiation*, where analytical derivatives are obtained manually or via computer algebra systems, and then implemented explicitly. Instead, AD relies on the observation that complex computations can often be split up into elementary steps, for which derivatives are readily implemented. If one can track those derivatives during the computation of the forward, or ‘primal’, function, the chain rule allows to obtain derivatives.

For this work, two properties of AD are particularly relevant: It allows the computation of derivatives with respect to quantities that are explicitly used in the forward computation, and it can do so at the same asymptotic computational cost as the forward function. In particular, AD can obtain two quantities efficiently: Given a differentiable function $u : \mathbb{R}^N \rightarrow \mathbb{R}^M$, the Jacobian of u is defined as the $M \times N$ matrix $\partial u_i(x)/\partial x_j$. AD can then obtain Jacobian-vector and vector-Jacobian products, i.e., the multiplication and summation of factors over either the input or the output dimension. This corresponds to propagating derivatives from the inputs *forwards*, leading to *forward-mode AD*, or from the end result *backwards*, leading to *reverse-mode AD*. As many popular AD frameworks are primarily implemented to work with neural networks, where scalar loss functions must be differentiated with respect to many parameters, reverse-mode AD, also called ‘backpropagation’,⁴⁷ is more generally available. More recent frameworks implement both approaches, for instance `jax`⁴³ which is used in the present work. AD can also be leveraged to compute contractions of higher-order derivative operators⁴⁸.

III. CONSTRUCTING GRAPH MLPs

This work considers periodic systems,⁴⁹ consisting of N atoms with atomic numbers Z_i placed in a simulation cell which is infinitely periodically tiled in space. We define

$$\begin{aligned}
 \mathbf{R}_{\text{sc}} &:= \{ \mathbf{r}_i : i = 1 \dots N \} && \text{positions in simulation cell} \\
 \mathbf{Z} &:= \{ Z_i : i = 1 \dots N \} && \text{atomic numbers} \\
 \mathbf{B} &:= \{ \mathbf{b}_a : a = 1, 2, 3 \} && \text{basis or lattice vectors} \\
 \mathbf{r}_{\text{in}} &:= \mathbf{r}_i + \sum_a n_a \mathbf{b}_a && \text{(replica) position} \\
 \mathbf{R}_{\text{all}} &:= \{ \mathbf{r}_{\text{in}} : \mathbf{r}_i \in \mathbf{R}_{\text{sc}}, \mathbf{n} \in \mathbb{Z}^3 \} && \text{all (bulk) positions} \\
 \mathbf{r}_{ij} &:= \mathbf{r}_j - \mathbf{r}_i && \text{atom-pair vector} \\
 |\mathbf{r}_{ij}^{\text{MIC}}| &:= \min_{\mathbf{n} \in \mathbb{Z}^3} | \mathbf{r}_j + \sum_a n_a \mathbf{b}_a - \mathbf{r}_i | && \text{minimum image convention}
 \end{aligned}$$

In this setting, a MLP is a function that maps the structure represented by its positions, lattice vectors, and atomic numbers, $(\mathbf{R}_{\text{sc}}, \mathbf{B}, \mathbf{Z})$, to a set of atomic potential energies $\mathbf{U} := \{ U_i : i = 1 \dots N \}$, which yield the total potential energy $U = \sum_{i=1} U_i$. Since AD relies on the forward computation to

calculate derivatives, it is sensitive to the exact implementation of this mapping. Care must therefore be taken to construct the MLP such that required derivatives are available.

This work considers MLPs that scale linearly with N . Therefore, the number of atoms contributing to a given U_i must be bounded, which is achieved by introducing a cutoff radius r_c , restricting interactions to finite-sized atomic neighborhoods $\mathbf{N}(i) = \{ \mathbf{r}_j : |\mathbf{r}_{ij}| \leq r_c, \mathbf{r}_j \in \mathbf{R}_{\text{all}} \}$. To ensure translational invariance, MLPs do not rely on neighbor positions directly, but rather on atom-pair vectors centered on i , from which atom-pair vectors between neighboring atoms can be constructed, for instance to determine angles.

The resulting structure can be seen as a graph G . The vertices V of this graph are identified with atoms, labeled with their respective atomic numbers, and connected by edges E that are labeled with atom-pair vectors if placed closer than r_c . Starting from G , MLPs can be constructed in different ways: Local MLPs compute a suitable representation⁵⁰ of each neighborhood, and predict U_i from that representation using a learned function, such as a neural network or a kernel machine. Such models are conceptually simple, but cannot account for effects that extend beyond r_c . Recently, semi-local models such as MPNNs^{22,29,51–58} have been introduced to tackle this shortcoming without compromising asymptotic runtime. In such models, effective longer-range interactions are built up iteratively by allowing adjacent neighborhoods to interact repeatedly. We introduce the parameter M , the *interaction depth*, to quantify how many such iterations are included. After M interactions, the energy at any given site can depend implicitly on positions within M hops on the graph, which we denote by $\mathbf{N}^M(i)$, leading to an *effective* cutoff radius $r_c^{\text{eff}} = M r_c$. However, since interactions are confined to neighborhoods at each iteration, the asymptotic linear scaling is not impacted. Local MLPs are formally included as semi-local models with $M=1$, allowing a unified treatment for both. We term this class of potentials, which act on sets of neighborhoods and use atom-pair vectors as input, *graph-based machine-learning potentials (GLPs)*. By construction, this framework does not include global interactions.

We consider two strategies to construct GLPs: The ‘standard’ way, which includes periodic boundary conditions via the edges in the graph, and an ‘unfolded’ formulation, where periodicity is explicitly included via replica positions.

In the standard architecture, vertices in G are identified with atoms *in the simulation cell*, using the minimum image convention (MIC) to include periodicity. Edges E_{ij} exist between two atoms i and j in \mathbf{R}_{sc} if they interact:

$$E_{ij}^{\text{std}} = \{ \mathbf{r}_{ij}^{\text{MIC}} : |\mathbf{r}_{ij}^{\text{MIC}}| \leq r_c \}. \quad (1)$$

We denote the the graph constructed in this manner as G^{std} and the set of edges E^{std} .

Alternatively, we can first determine the total set of positions $\mathbf{R}_{\text{unf}} \subset \mathbf{R}_{\text{all}}$ that can interact with atoms in the simulation cell, creating an ‘unfolded’ system extracted from the bulk, consisting of \mathbf{R}_{sc} and all replicas with up to r_c^{eff} distance from the cell boundary. This construction can be performed efficiently, and adds only a number of positions that is proportional to the *surface* of the simulation cell, therefore becoming

increasingly negligible as N increases at constant density⁴². We proceed by constructing a correspondingly modified graph G^{unf} , and compute potential energies for vertices corresponding to atoms in the simulation cell only. By construction, since the same atom-pair vectors appear in the graph, this approach then reproduces the potential energy of the standard method.

IV. DERIVATIVES

Having constructed the forward function for a given GLP, we can compute derivatives with respect to its inputs using AD. In this section, we discuss how forces, stress, and heat flux can be computed in this manner, and demonstrate the relationship between different formulations.

A. Forces

For MD, the most relevant quantities are the forces

$$\mathbf{F}_i = -\frac{\partial U}{\partial \mathbf{r}_i} \quad (2)$$

acting on the atoms in the simulation cell. Since \mathbf{R}_{sc} are an explicit input, they can be computed directly with AD: U is a scalar, and this therefore is a trivial Jacobian-vector product, which can be computed with the same asymptotic cost as U .

An interesting situation arises if pairwise forces are desired. Strictly speaking, in a many-body MLP, where interactions cannot be decomposed into pairwise contributions, such quantities are not well-defined and Newton's third law is replaced by conservation of momentum, which requires $\sum_{i=1}^N \mathbf{F}_i = 0$. Nevertheless, pairwise forces with an antisymmetric structure can be defined by exploiting the construction of GLPs in terms of atom-pair vectors. In the standard formulation, U is a function of all edges,

$$U = U(\{\mathbf{r}_{ij} : ij \in \mathbf{E}^{\text{std}}\}). \quad (3)$$

Hence, by the chain rule,

$$\mathbf{F}_i = \sum_{j \in \mathbf{N}(i)} \frac{\partial U}{\partial \mathbf{r}_{ij}} - \frac{\partial U}{\partial \mathbf{r}_{ji}} \quad (4)$$

$$=: \sum_{j \in \mathbf{N}(i)} \mathbf{F}_{ij}. \quad (5)$$

The pairwise forces such defined exhibit anti-symmetry, and therefore fulfil Newton's third law. For $M=1$, the local case, this definition reduces to a more standard form³⁷

$$\mathbf{F}_{ij} = \frac{\partial U_i}{\partial \mathbf{r}_{ij}} - \frac{\partial U_j}{\partial \mathbf{r}_{ji}}. \quad (6)$$

However, for general GLPs with $M > 1$, this definition includes a sum over all U_k that are influenced by a given edge

$$\mathbf{F}_{ij} = \sum_{k \in \mathbf{N}^M(i)} \frac{\partial U_k}{\partial \mathbf{r}_{ij}} - \frac{\partial U_k}{\partial \mathbf{r}_{ji}}, \quad (7)$$

subverting expectations connecting local potential energies to pairwise forces. We note that this seeming contradiction is a consequence of the combination of the peculiar construction of GLPs and AD: In principle, it is always possible to define extended neighborhoods up to r_c^{eff} , obtaining U_i purely as a function of atom-pair vector originating from i . However, to construct derivatives with respect to these atom-pair vectors with AD, these extended neighborhoods have to be constructed and included explicitly, therefore negating the computational efficiency gains of a GLP architecture.

B. Stress

The definition of the (potential) stress is⁵⁹

$$\boldsymbol{\sigma} = \frac{1}{V} \frac{\partial U}{\partial \boldsymbol{\epsilon}} \Big|_{\boldsymbol{\epsilon}=0}, \quad (8)$$

with U denoting the potential energy after a strain transformation with the the 3×3 symmetric tensor $\boldsymbol{\epsilon}$

$$\mathbf{r} \rightarrow (\mathbb{1} + \boldsymbol{\epsilon}) \cdot \mathbf{r}, \quad (9)$$

acting on \mathbf{R}_{all} . While computing this derivative for arbitrary potentials and periodic systems has required 'much effort'³¹ in the past^{33,60}, it is straightforward with AD.

The simplest approach, followed for instance by `schneppack`^{61,62} and `nequip`^{54,63}, is to inject the strain transformation explicitly into the construction of the GLP. This can be done at different points: One can transform \mathbf{R}_{sc} and \mathbf{B} *before* constructing \mathbf{G} , directly transform atom-pair vectors, or transform all contributing positions \mathbf{R}_{unf} . Alternatively, as the inner derivative of inputs with respect to $\boldsymbol{\epsilon}$ is simply the input, the derivative of U with respect to inputs can be obtained with AD, and the stress computed analytically from the results. This avoids modifying the forward computation of U entirely.

These approaches yield, with \otimes denoting an outer product,

$$V \boldsymbol{\sigma} = \frac{\partial U(\mathbf{R}_{\text{sc}}, \mathbf{B})}{\partial \boldsymbol{\epsilon}} \quad (10)$$

$$= \frac{\partial U(\mathbf{E})}{\partial \boldsymbol{\epsilon}} \quad (11)$$

$$= \frac{\partial U(\mathbf{R}_{\text{unf}})}{\partial \boldsymbol{\epsilon}} \quad (12)$$

$$= \sum_{i \in \mathbf{R}_{\text{sc}}} \mathbf{r}_i \otimes \frac{\partial U}{\partial \mathbf{r}_i} + \sum_{b \in \mathbf{B}} \mathbf{b} \otimes \frac{\partial U}{\partial \mathbf{b}} \quad (13)$$

$$= \sum_{ij \in \mathbf{E}} \mathbf{r}_{ij} \otimes \frac{\partial U}{\partial \mathbf{r}_{ij}} \quad (14)$$

$$= \sum_{i \in \mathbf{R}_{\text{unf}}} \mathbf{r}_i \otimes \frac{\partial U}{\partial \mathbf{r}_i}, \quad (15)$$

recovering previous formulations given by Louwse and Baerends³¹ and Thompson³³. As seen in Tables I and III, all such forms of the stress are equivalent, provided the strain transformation is applied consistently to all used inputs.⁶⁴

In all cases, as U is differentiated with respect to its inputs, asymptotic cost remains linear.

A more complex situation arises if strain derivatives of atomic energies, i.e., atomic stresses

$$\sigma_i := \frac{1}{V} \frac{\partial U_i}{\partial \epsilon} \quad i \in \mathbf{R}_{\text{sc}} \quad (16)$$

are required. Their calculation requires either one backward pass per U_i , or one forward pass for each entry in ϵ . If only reverse-mode AD is available, its evaluation therefore scales quadratically with N . Linear scaling is retained with forward mode. For GLPs with $M=1$, linear scaling in reverse mode can be recovered by using Eq. (14): Every edge can be uniquely assigned to one U_i , and therefore the derivatives can be used to construct atomic stresses. For $M>1$, this is not possible; similar to the observations of the previous section, atomic stresses take a semi-local form.

C. Heat Flux

Finally, we discuss the heat flux, which is required to compute thermal conductivities with the Green-Kubo (GK) method^{65–67}. It describes how energy flows between atoms, and has been the focus of a large body of previous work^{32,35,37,40–42,60}.

The fundamental definition of the heat flux for MLPs was originally derived by Hardy⁶⁸ for periodic quantum systems. It reads⁴²

$$\mathbf{J}_{\text{full}} = \sum_{\substack{i \in \mathbf{R}_{\text{sc}} \\ j \in \mathbf{R}_{\text{all}}}} \left(\mathbf{r}_{ji} \left(\frac{\partial U_i}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j \right) \right) + \sum_{i \in \mathbf{R}_{\text{sc}}} E_i \mathbf{v}_i \quad (17)$$

$$=: \mathbf{J}_{\text{pot}} + \mathbf{J}_{\text{conv}}, \quad (18)$$

where \mathbf{v}_i denote velocities, m_i masses, and $E_i = U_i + 1/2m_i\mathbf{v}_i^2$ is the total energy per atom. Intuitively, the ‘potential’ term \mathbf{J}_{pot} describes how the total instantaneous change in U_i can be attributed to interactions with other atoms, with energy flowing between them, while the second, ‘convective’, term \mathbf{J}_{conv} describes energy being carried the individual atoms. In the present setting, \mathbf{J}_{conv} can be computed directly, as E_i are available. \mathbf{J}_{pot} , however, presents a challenge in an AD framework: In principle, \mathbf{J}_{pot} could be computed directly, obtaining the required partial derivatives with AD. However, as \mathbf{J}_{pot} is neither a Jacobian-vector nor a vector-Jacobian product, this requires repeated evaluations over the input or output dimension. Even when restricting $j \in \mathbf{R}_{\text{sc}}$, which can be achieved by introducing the MIC for \mathbf{r}_{ji} (see Supp. Mat. for details),

$$\mathbf{J}_{\text{pot}}^{\text{MIC}} = \sum_{i,j \in \mathbf{R}_{\text{sc}}} \left(\mathbf{r}_{ji}^{\text{MIC}} \left(\frac{\partial U_i}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j \right) \right), \quad (19)$$

computational cost of a direct implementation with AD scales quadratically with N , rendering the system sizes and simulation times required for the GK method inaccessible⁴². We therefore consider approaches that restore linear scaling in the following.

For $M=1$, edges can be uniquely assigned to atomic energy contributions as discussed for atomic stresses in Sec. IV B. In this case

$$\frac{\partial U}{\partial \mathbf{r}_{ij}} = \sum_{k \in \mathbf{R}_{\text{sc}}} \frac{\partial U_k}{\partial \mathbf{r}_{ij}} = \frac{\partial U_i}{\partial \mathbf{r}_{ij}} = \frac{\partial U_i}{\partial \mathbf{r}_j}, \quad (20)$$

so that

$$\mathbf{J}_{\text{pot}}^{M=1} = \sum_{ij \in \mathbf{E}} \left(\mathbf{r}_{ji} \left(\frac{\partial U}{\partial \mathbf{r}_{ij}} \cdot \mathbf{v}_j \right) \right), \quad (21)$$

which requires a single evaluation of reverse-mode AD.

We note that the terms appearing in front of \mathbf{v}_j also appear in the stress in Eq. (14). However, for a given j , the pre-factor *cannot* be identified with the atomic stress as defined in Eq. (16) – the atomic energy being differentiated is not U_j , but U_i . The indices can only be exchanged for additive pairwise potentials; this inequivalence was recently corrected in the LAMMPS code⁴⁰.

This approach is not applicable for $M>1$, since the relation in Eq. (20) no longer holds, and the mapping between stress contributions and heat flux contributions becomes invalid. By using the unfolded construction, however, linear scaling can be restored regardless⁴². Introducing auxiliary positions $\mathbf{r}_i^{\text{aux}}$, which are numerically identical to the positions \mathbf{r}_i , but not used to compute U , and defining the energy barycenter $\mathbf{B} = \sum_{i \in \mathbf{R}_{\text{sc}}} \mathbf{r}_i^{\text{aux}} U_i$, the heat flux can be written as

$$\mathbf{J}_{\text{pot}}^{M \geq 1} = \sum_{j \in \mathbf{R}_{\text{unf}}} \frac{\partial \mathbf{B}}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j - \sum_{j \in \mathbf{R}_{\text{unf}}} \left(\mathbf{r}_j \left(\frac{\partial U}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j \right) \right). \quad (22)$$

The first term requires three reverse-mode evaluations, or one forward-mode evaluation, the latter a single backward- or forward-mode evaluation. Since the overhead introduced by explicitly constructing \mathbf{R}_{unf} scales as $N^{2/3}$, overall linear scaling is restored, albeit with a pre-factor due to the higher number of positions to be considered.

To summarize, we have introduced two forms of the heat flux that can be implemented efficiently with AD: Equation (21), which applies to GLPs with $M=1$, and Eq. (22), which applies for $M \geq 1$, but introduces some additional overhead. Both are equivalent to the general, quadratically-scaling, form given in Eq. (17), as seen in Tables II and IV.

V. EXPERIMENTS

A. Lennard-Jones Argon

The stress formulas in Eqs. (10) to (15) and heat flux formulas Eqs. (19), (21) and (22) have been implemented in the `glp` package⁴⁴ using `jax`⁴³. As a first step, we numerically verify this implementation.

To this end, the Lennard-Jones potential⁴⁵ is employed, where analytical derivatives including those required for the heat flux are readily available, and implementations are included in many packages, for example the Atomic Simulation

Equation	Single		Double	
	MAE (eV)	MAPE (%)	MAE (eV)	MAPE (%)
Fin. diff.	$7.70 \cdot 10^{-4}$	$1.04 \cdot 10^{-1}$	$1.13 \cdot 10^{-6}$	$1.18 \cdot 10^{-4}$
10	$1.19 \cdot 10^{-5}$	$1.79 \cdot 10^{-3}$	$3.15 \cdot 10^{-6}$	$3.69 \cdot 10^{-4}$
11	$8.25 \cdot 10^{-6}$	$1.27 \cdot 10^{-3}$	$3.15 \cdot 10^{-6}$	$3.69 \cdot 10^{-4}$
12	$9.17 \cdot 10^{-6}$	$1.36 \cdot 10^{-3}$	$3.15 \cdot 10^{-6}$	$3.69 \cdot 10^{-4}$
13	$1.18 \cdot 10^{-5}$	$1.79 \cdot 10^{-3}$	$3.15 \cdot 10^{-6}$	$3.69 \cdot 10^{-4}$
14	$8.22 \cdot 10^{-6}$	$1.27 \cdot 10^{-3}$	$3.15 \cdot 10^{-6}$	$3.69 \cdot 10^{-4}$
15	$9.16 \cdot 10^{-6}$	$1.37 \cdot 10^{-3}$	$3.15 \cdot 10^{-6}$	$3.69 \cdot 10^{-4}$

TABLE I. Error in stress for Lennard-Jones argon, comparing different formulations, as well as finite differences, to analytical derivatives. Results are shown for both single and double precision arithmetic, and for $\sigma \cdot V$ in place of σ .

Eq.	Single		Double	
	MAE (eVÅ/fs)	MAPE (%)	MAE (eVÅ/fs)	MAPE (%)
19	$2.81 \cdot 10^{-9}$	$1.71 \cdot 10^{-2}$	$1.47 \cdot 10^{-10}$	$6.81 \cdot 10^{-4}$
21	$2.84 \cdot 10^{-9}$	$1.67 \cdot 10^{-2}$	$1.47 \cdot 10^{-10}$	$6.81 \cdot 10^{-4}$
22	$2.44 \cdot 10^{-9}$	$1.54 \cdot 10^{-2}$	$1.47 \cdot 10^{-10}$	$6.81 \cdot 10^{-4}$

TABLE II. Error in heat flux for Lennard-Jones argon, comparing different formulations to analytical derivatives. Results are shown for both single and double precision arithmetic.

Environment (ASE)⁶⁹. In the GLP framework, the Lennard-Jones potential can be seen as an extreme case of a $M=1$ GLP, where U_i is composed of a sum of pair terms:

$$U_i = \frac{1}{2} \sum_{j \in \mathcal{N}(i)} 4\epsilon \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right). \quad (23)$$

For this experiment, parameters approximating elemental argon are used⁷⁰. 100 randomly displaced and distorted geometries, based on the 512-atom $8 \times 8 \times 8$ supercell of the face-centered cubic primitive cell with lattice parameter 3.72 Å and angle 60° are used. Random velocities to evaluate a finite heat flux are sampled from the Boltzmann distribution corresponding to 10 K. Additional computational details are discussed in the Supp. Mat.

Table I compares the stress formulations in Eqs. (10) to (15) with finite differences. We report ‘best-case’ results for finite differences, choosing the stepsize that minimises the error. In the table, the mean absolute error (MAE) and mean absolute percentage error (MAPE) with respect to the analytical ground truth are reported. All given formulations are found to be equivalent. In single precision arithmetic, the AD-based implementations slightly outperform finite differences, in double precision, errors are similar.

For the heat flux, finite difference approaches are not feasible. Therefore, only AD-based implementations are shown in Table II. In the case of the Lennard-Jones potential, where $M=1$, Eqs. (19), (21) and (22) are found to be identical.

Equation	Single		Double	
	MAE (eV)	MAPE (%)	MAE (eV)	MAPE (%)
10	$1.58 \cdot 10^{-2}$	$4.40 \cdot 10^{-2}$	$1.45 \cdot 10^{-4}$	$2.32 \cdot 10^{-4}$
11	$1.58 \cdot 10^{-2}$	$4.38 \cdot 10^{-2}$	$1.45 \cdot 10^{-4}$	$2.32 \cdot 10^{-4}$
12	$1.57 \cdot 10^{-2}$	$4.33 \cdot 10^{-2}$	$1.45 \cdot 10^{-4}$	$2.32 \cdot 10^{-4}$
13	$1.58 \cdot 10^{-2}$	$4.40 \cdot 10^{-2}$	$1.45 \cdot 10^{-4}$	$2.32 \cdot 10^{-4}$
14	$1.58 \cdot 10^{-2}$	$4.38 \cdot 10^{-2}$	$1.45 \cdot 10^{-4}$	$2.32 \cdot 10^{-4}$
15	$1.57 \cdot 10^{-2}$	$4.33 \cdot 10^{-2}$	$1.45 \cdot 10^{-4}$	$2.32 \cdot 10^{-4}$

TABLE III. Error in stress for tin selenide, comparing different formulations to finite differences, for SO3KRATES with $M=2$. Results are shown for both single and double precision arithmetic, and for $\sigma \cdot V$ in place of σ . Results for other M , which are similar to the one shown here, can be found in the Supp. Mat.

Eq.	M	Single		Double	
		MAE (eVÅ/fs)	MAPE (%)	MAE (eVÅ/fs)	MAPE (%)
21	1	$5.78 \cdot 10^{-9}$	$9.74 \cdot 10^{-4}$	$1.09 \cdot 10^{-17}$	$1.73 \cdot 10^{-12}$
22	1	$8.75 \cdot 10^{-8}$	$2.65 \cdot 10^{-2}$	$1.69 \cdot 10^{-16}$	$4.31 \cdot 10^{-11}$
21	2	$3.73 \cdot 10^{-3}$	$5.84 \cdot 10^2$	$3.73 \cdot 10^{-3}$	$5.84 \cdot 10^2$
22	2	$9.36 \cdot 10^{-8}$	$1.00 \cdot 10^{-2}$	$1.54 \cdot 10^{-16}$	$1.60 \cdot 10^{-11}$
21	3	$1.01 \cdot 10^{-3}$	$3.27 \cdot 10^2$	$1.01 \cdot 10^{-3}$	$3.25 \cdot 10^2$
22	3	$9.74 \cdot 10^{-8}$	$3.04 \cdot 10^{-2}$	$1.65 \cdot 10^{-16}$	$2.91 \cdot 10^{-11}$

TABLE IV. Error in heat flux for tin selenide, comparing different formulations to the baseline implementation in Eq. (19) for SO3KRATES models with differing numbers of interaction steps M . Results are shown for both single and double precision arithmetic.

VI. TIN SELENIDE WITH SO3KRATES

To investigate stress and heat flux in a practical setting, we now study tin selenide (SnSe) using the state-of-the-art SO3KRATES GLP⁴⁶. In contrast to other equivariant MLPs, for instance NEQUIP⁵⁴, SO3KRATES replaces shared equivariant feature representations by separated branches for invariant and equivariant information, whose information exchange is handled using an equivariant self-attention mechanism. By doing so, one can achieve data efficiency and extrapolation quality competitive to state-of-the-art GLPs at reduced time and memory complexity. As non-local interactions are not modeled in the GLP framework introduced in this work, global interactions are disabled in the SO3KRATES models used at present.

For these experiments, SO3KRATES models with $M=1, 2, 3$ were trained on approximately 3000 reference calculations, comprising a number of thermalization trajectories at different volumes at 300 K. These calculations were performed as part of a large-scale *ab initio* Green-Kubo (aiGK) benchmark study by Knoop et al.^{71,72}. Additional details on the MLP training can be found in the Supp. Mat.

	DFT	$M=1$	$M=2$	$M=3$
B_0 (eV/Å ³)	0.230	0.236	0.229	0.223
B_p (eV/Å ⁶)	5.576	2.830	4.815	7.118
V_0 (Å ³)	26.429	26.489	26.322	26.388
Error B_0 (%)	–	2.58	–0.47	–3.14
Error B_p (%)	–	–49.25	–13.64	27.66
Error V_0 (%)	–	0.22	–0.40	–0.16

TABLE V. Cohesive properties of SnSe for PBEsol-DFT, and SO3KRATES with different numbers of interaction steps M obtained via the Vinet EOS. Best values are highlighted. Values are in good agreement with existing literature⁷⁷.

1. Implementation of stress and heat flux

While no analytical derivatives are available for SO3KRATES, the implementation of the stress can be verified with finite differences, and the heat flux can be checked for consistency between different implementations. Similar to Sec. V A, we use 100 randomly displaced and distorted $4 \times 8 \times 8$ supercells of the 0 K primitive cell of SnSe for this experiment, sampling velocities from the Boltzmann distribution at 10 K to evaluate a finite heat flux.

Table III compares the stress implementations in Eqs. (10) to (15) with finite differences, confirming the equivalence of all implemented formulations.

Table IV compares the heat flux formulations in Eq. (21) and Eq. (22) with the baseline in Eq. (19), implementing the quadratically-scaling ‘Hardy’ heat flux with the MIC. For $M=1$, all formulations are precisely equivalent. For $M>1$, the semi-local case, only Eq. (22) is equivalent to the ‘Hardy’ heat flux; Eq. (21) does not apply and consequently is not equivalent, displaying large deviations.

2. Equation of state and pressure

To assess the capability of SO3KRATES to predict stress- and pressure-related materials properties, we calculate energy-volume and pressure-volume curves for SnSe to obtain an equation of state (EOS) of the Vinet form^{73,74}. The experiment is performed for unit cells that were homogeneously strained up to $\pm 2\%$ starting from the fully relaxed geometry, and relaxing the internal degrees of freedom afterwards. The energy vs. volume curves for SO3KRATES with $M=1,2,3$ interaction steps are shown in Fig. 1 in comparison to the DFT reference using the PBEsol exchange-correlation functional with ‘light’ default basis sets in FHI-aims^{75,76}. SO3KRATES with three interaction steps ($M=3$) yields the best visual agreement for the energy-volume curve in Fig. 1, and the best equilibrium volume. To quantify the agreement, we evaluate the Vinet EOS and extract the cohesive properties equilibrium volume V_0 , isothermal Bulk modulus B_0 , and its pressure derivative B'_0 (functional forms are given in the Supp. Mat.). Results are listed in Table V. For SO3KRATES with three interaction steps ($M=3$), the predicted

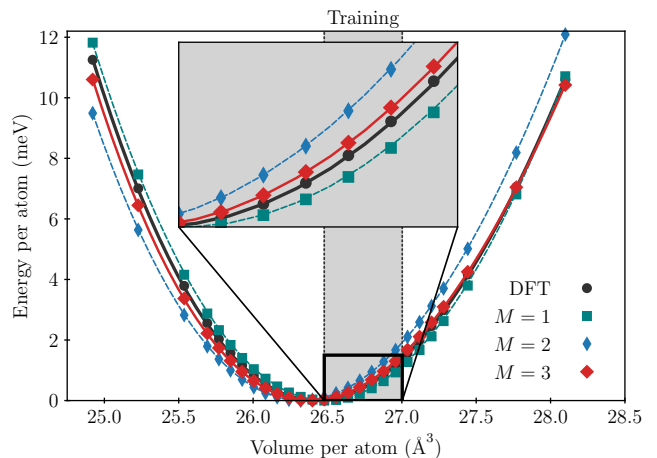


FIG. 1. EOS (energy vs. volume) computed with PBEsol-DFT (black dots) compared to SO3KRATES with different numbers of interaction steps M . The connecting lines have been obtained by fitting the Vinet EOS. Inset: Zoom into the region of volumes that were covered during the training indicated by the gray shading.

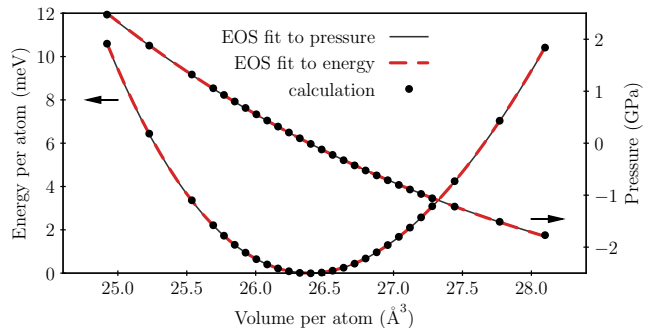


FIG. 2. Comparison of EOS obtained by fitting energy-volume (red dashed) and pressure-volume (black solid) data. The results are in perfect agreement.

volume deviates by -0.16% from the DFT reference, and the deviation of the bulk modulus is -3.14% . A larger error is seen for the pressure derivative of the bulk modulus, B'_0 , which deviates by 27.7% , indicating worse agreement further away from the training region. Overall, the agreement between DFT and SO3KRATES when predicting cohesive properties can be considered satisfactory. The energy-volume predictions are very good, and transfer even to volumes that are larger or smaller than the ones seen during training.

Finally we check the internal consistency of energy and stress predictions with SO3KRATES by fitting energy-volume and pressure-volume curves and verify that they yield identical parameters for the EOS. Results are shown in Fig. 2. As seen there, results are in perfect agreement, which verifies that stress and resulting pressure are consistent with the underlying energy function as expressed in Eq. (8).

Source	Method	κ (W/mK)	κ_x	κ_y	κ_z
This work	SO3KRATES, $M=1$	0.99 ± 0.10	0.53 ± 0.03	1.31 ± 0.13	1.12 ± 0.12
"	SO3KRATES, $M=2$	1.13 ± 0.07	0.48 ± 0.04	1.59 ± 0.07	1.20 ± 0.07
"	SO3KRATES, $M=3$	1.13 ± 0.10	0.56 ± 0.05	1.56 ± 0.15	1.32 ± 0.16
Brorsson et al. ⁷⁸	FCP	1.12	0.57	1.46	1.32
Liu et al. ⁷⁹	MLP	0.86 ± 0.13	0.57 ± 0.05	1.25 ± 0.24	0.76 ± 0.08
Knoop et al. ⁷¹	DFT (extrapolated)	1.40 ± 0.39	–	–	–
Review by Wei et al. ⁸⁰	Experiments	0.45 to 1.9	–	–	–

TABLE VI. Thermal conductivity κ of tin selenide at 300 K.

3. Thermal Conductivity

Finally, we proceed to GK calculations, following the approach outlined in our previous work^{42,81}. Finding that simulation cells with $N=864$ atoms and a simulation duration of $t_0=1$ ns yield converged results (see Supp. Mat.), we run 11 MD simulations using the `glp` package and SO3KRATES, computing \mathbf{J} at every step.

Table VI shows the result: For $M=2,3$, SO3KRATES in excellent agreement with the work by Brorsson et al.⁷⁸, which uses a force constant potential (FCP). A larger difference is observed with the work by Liu et al.⁷⁹, who, however, use the PBE exchange-correlation functional, as opposed to PBEsol, which was used for the present work. The observed thermal conductivity is consistent with experiments, as well as the size-extrapolated DFT result of Knoop et al.⁷¹. The anisotropy of κ in SnSe is captured as well. Overall, SO3KRATES with more than one interaction step is able to capture the converged thermal conductivity of SnSe, using only training data from the thermalization step of a full aiGK workflow; long-running equilibrium *ab initio* MD simulations, the bottleneck of the GK method, have been avoided.

VII. CONCLUSION

We demonstrated that the stress and heat flux can be computed efficiently with AD for potentials based on a graph of atom-pair vectors, which we termed GLPs, and provided example implementations in the `glp` package⁴⁴. Numerical experiments for Lennard-Jones argon and tin selenide with the SO3KRATES GLP, verified that these quantities are computed correctly and consistently. The equivariant SO3KRATES GLP was shown to predict cohesive properties and thermal conductivity of SnSe in good agreement with DFT, other MLPs, and experiments, confirming the practical relevance of computational access to stress and heat flux.

This work enables the use of a large class of recently developed MLPs, those which can be described in the GLP framework, in computational materials science, and in particular for the calculation of thermal conductivities using the GK method. For GLPs implemented with `jax`, the `glp` package is provided to enable the calculation of stress and heat flux without requiring further implementation efforts.

DATA AND CODE AVAILABILITY

The data and code that support the findings of this study are available at doi:10.5281/zenodo.7852530 and at <https://github.com/sirmarcel/glp-archive>. The `glp` package is available at <https://github.com/sirmarcel/glp>. The SO3KRATES model is implemented in `mlff`, available at <https://github.com/thorben-frank/mlff>. Further information can be found in the Supp. Mat. and at <https://marcel.science/glp>.

ACKNOWLEDGMENTS

M.F.L. was supported by the German Ministry for Education and Research BIFOLD program (refs. 01IS18025A and 01IS18037A), and by the TEC1p Project (ERC Horizon 2020 No. 740233). M.F.L. would like to thank Samuel Schoenholz and Niklas Schmitz for constructive discussions, Shuo Zhao for feedback on the manuscript, and acknowledges contributions by Fabian Nagel and Adam Norris. J.T.F acknowledges support from the Federal Ministry of Education and Research (BMBF) and BIFOLD program (refs. 01IS18025A and 01IS18037A). F.K. acknowledges support from the Swedish Research Council (VR) program 2020-04630, and the Swedish e-Science Research Centre (SeRC). The computations were partially enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre (NSC) and by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at NSC partially funded by the Swedish Research Council through grant agreement no. 2022-06725. Xuan Gu at NSC is acknowledged for technical assistance on the Berzelius resource.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Marcel F. Langer: Conceptualization; Writing – original draft (lead); Writing – review & editing (lead); Software (lead); Visualization (equal); Investigation (equal); Project administration. **J. Thorben Frank:** Writing – original draft (supporting); Writing – review & editing (supporting); Software (supporting); Investigation (equal). **Florian Knoop:** Writing – original draft (supporting); Writing – review & editing (supporting); Visualization (equal); Investigation (equal).

- ¹M. E. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation* (Oxford University Press, 2010).
- ²R. Car and M. Parrinello, “Unified approach for molecular dynamics and density-functional theory,” *Phys. Rev. Lett.* **55**, 2471–2474 (1985).
- ³A. M. Teale, T. Helgaker, A. Savin, C. Adamo, B. Aradi, A. V. Arbuznikov, P. W. Ayers, E. J. Baerends, V. Barone, P. Calaminici, E. Cancès, E. A. Carter, P. K. Chattaraj, H. Chermette, I. Ciofini, T. D. Crawford, F. De Proft, J. F. Dobson, C. Draxl, T. Frauenheim, E. Fromager, P. Fuentealba, L. Gagliardi, G. Galli, J. Gao, P. Geerlings, N. Gidopoulos, P. M. W. Gill, P. Gori-Giorgi, A. Görling, T. Gould, S. Grimme, O. Gritsenko, H. J. A. Jensen, E. R. Johnson, R. O. Jones, M. Kaupp, A. M. Köster, L. Kronik, A. I. Krylov, S. Kvaal, A. Laestadius, M. Levy, M. Lewin, S. Liu, P.-F. Loos, N. T. Maitra, F. Neese, J. P. Perdew, K. Pernal, P. Pernot, P. Piecuch, E. Rebolini, L. Reining, P. Romaniello, A. Ruzsinszky, D. R. Salahub, M. Scheffler, P. Schwerdtfeger, V. N. Staroverov, J. Sun, E. Tellgren, D. J. Tozer, S. B. Trickey, C. A. Ullrich, A. Vela, G. Vignale, T. A. Wesolowski, X. Xu, and W. Yang, “Dft exchange: sharing perspectives on the workhorse of quantum chemistry and materials science,” *Phys. Chem. Chem. Phys.* **24**, 28700–28781 (2022).
- ⁴M. González, “Force fields and molecular dynamics simulations,” *J. Neutron.* **12**, 169–200 (2011).
- ⁵S. Lorenz, A. Groß, and M. Scheffler, “Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks,” *Chem. Phys. Lett.* **395**, 210–215 (2004).
- ⁶G. Li, J. Hu, S.-W. Wang, P. G. Georgopoulos, J. Schoendorf, and H. Rabitz, “Random sampling-high dimensional model representation (RS-HDMR) and orthogonality of its different order component functions,” *J. Phys. Chem. A* **110**, 2474–2485 (2006).
- ⁷J. Behler and M. Parrinello, “Generalized neural-network representation of high-dimensional potential-energy surfaces,” *Phys. Rev. Lett.* **98**, 146401 (2007).
- ⁸A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, “Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons,” *Phys. Rev. Lett.* **104**, 136403 (2010).
- ⁹S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, “Towards exact molecular dynamics simulations with machine-learned force fields,” *Nat. Commun.* **9**, 3887 (2018).
- ¹⁰S. Chmiela, H. E. Sauceda, I. Poltavsky, K.-R. Müller, and A. Tkatchenko, “sGDML: Constructing accurate and data efficient molecular force fields using machine learning,” *Comput. Phys. Comm.* **240**, 38–45 (2019).
- ¹¹I. Poltavsky and A. Tkatchenko, “Machine learning force fields: Recent advances and remaining challenges,” *J. Phys. Chem. Lett.* **12**, 6551–6564 (2021).
- ¹²F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, “Machine learning for molecular simulation,” *Annu. Rev. Phys. Chem.* **71**, 361–390 (2020).
- ¹³O. T. Unke, D. Koner, S. Patra, S. Käser, and M. Meuwly, “High-dimensional potential energy surfaces for molecular simulations: From empiricism to machine learning,” *Mach. Learn. Sci. Tech.* **1**, 013001 (2020).
- ¹⁴J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, and A. Tkatchenko, “Combining machine learning and computational chemistry for predictive insights into chemical systems,” *Chem. Rev.* **121**, 9816–9872 (2021).
- ¹⁵O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, “Machine learning force fields,” *Chem. Rev.* **121**, 10142–10186 (2021).
- ¹⁶O. T. Unke, M. Stöhr, S. Ganscha, T. Unterthiner, H. Maennel, S. Kashubin, D. Ahlin, M. Gastegger, L. Medrano Sandonas, A. Tkatchenko, and K.-R. Müller, “Accurate machine learned quantum-mechanical force fields for biomolecular simulations,” (2022), arXiv:2205.08306.
- ¹⁷G. Csányi, T. Albaret, M. C. Payne, and A. De Vita, ““learn on the fly”: A hybrid classical and quantum-mechanical molecular dynamics simulation,” *Phys. Rev. Lett.* **93**, 175503 (2004).
- ¹⁸R. Jinnouchi, K. Miwa, F. Karsai, G. Kresse, and R. Asahi, “On-the-fly active learning of interatomic potentials for large-scale atomistic simulations,” *J. Phys. Chem. Lett.* **11**, 6946–6955 (2020).
- ¹⁹C. Verdi, F. Karsai, P. Liu, R. Jinnouchi, and G. Kresse, “Thermal transport and phase transitions of zirconia by on-the-fly machine-learned interatomic potentials,” *NPJ Comput. Mater.* **7**, 156 (2021).
- ²⁰C. van der Oord, M. Sachs, D. P. Kovács, C. Ortner, and G. Csányi, “Hyperactive learning (hal) for data-driven interatomic potentials,” (2022), arXiv:2210.04225.
- ²¹Y. Xie, J. Vandermause, S. Ramakers, N. H. Protik, A. Johansson, and B. Kozinsky, “Uncertainty-aware molecular dynamics from bayesian active learning: Phase transformations and thermal transport in SiC,” (2022), arXiv:2203.03824.
- ²²J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, Australia, August 6–11* (PMLR, 2017) pp. 1263–1272.
- ²³N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, “Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds,” in *NeurIPS 2018 Workshop on Machine Learning for Molecules and Materials, Montréal, Canada, December 8* (Curran Assoc., 2018).
- ²⁴R. Drautz, “Atomic cluster expansion for accurate and transferable interatomic potentials,” *Phys. Rev. B* **99**, 249901 (2019).
- ²⁵A. Griewank and A. Walther, *Evaluating Derivatives* (Society for Industrial and Applied Mathematics, 2008).
- ²⁶A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, “Automatic differentiation in machine learning: A survey,” *J. Mach. Learn. Res.* **18**, 5595–5637 (2017).
- ²⁷S. Schoenholz and E. D. Cubuk, “Jax md: A framework for differentiable physics,” in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020), virtual, December 6–12* (Curran Assoc., 2020) pp. 11428–11441, not available.
- ²⁸K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller, “SchNetPack: A deep learning toolbox for atomistic systems,” *J. Chem. Theor. Comput.* **15**, 448–455 (2018).
- ²⁹O. T. Unke and M. Meuwly, “PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges,” *J. Chem. Theor. Comput.* **15**, 3678–3693 (2019).
- ³⁰K. T. Schütt, S. S. P. Hessmann, N. W. A. Gebauer, J. Lederer, and M. Gastegger, “Schnetpack 2.0: A neural network toolbox for atomistic machine learning,” *J. Chem. Phys.* **158**, 144801 (2023).
- ³¹M. J. Louwerse and E. J. Baerends, “Calculation of pressure in case of periodic boundary conditions,” *Chem. Phys. Lett.* **421**, 138–141 (2006).
- ³²D. Torii, T. Nakano, and T. Ohara, “Contribution of inter- and intramolecular energy transfers to heat conduction in liquids,” *J. Chem. Phys.* **128**, 044504 (2008).
- ³³A. P. Thompson, S. J. Plimpton, and W. Mattson, “General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions,” *J. Chem. Phys.* **131**, 154107 (2009).
- ³⁴S. G. Volz and G. Chen, “Molecular dynamics simulation of thermal conductivity of silicon nanowires,” *Appl. Phys. Lett.* **75**, 2056–2058 (1999).
- ³⁵Y. Chen, “Local stress and heat flux in atomistic systems involving three-body forces,” *J. Chem. Phys.* **124**, 054113 (2006).
- ³⁶A. Guajardo-Cuéllar, D. B. Go, and M. Sen, “Evaluation of heat current formulations for equilibrium molecular dynamics calculations of thermal conductivity,” *J. Chem. Phys.* **132**, 104111 (2010).
- ³⁷Z. Fan, L. F. C. Pereira, H.-Q. Wang, J.-C. Zheng, D. Donadio, and A. Harju, “Force and heat current formulas for many-body potentials in molecular dynamics simulations with applications to thermal conductivity calculations,” *Phys. Rev. B* **92**, 094301 (2015).
- ³⁸A. Marcolongo, P. Umari, and S. Baroni, “Microscopic theory and quantum simulation of atomic heat transport,” *Nat. Phys.* **12**, 80–84 (2015).
- ³⁹C. Carbogno, R. Ramprasad, and M. Scheffler, “Ab initio Green-Kubo approach for the thermal conductivity of solids,” *Phys. Rev. Lett.* **118**, 175901 (2017).

- (2017).
- ⁴⁰P. Boone, H. Babaei, and C. E. Wilmer, “Heat flux for many-body interactions: Corrections to LAMMPS,” *J. Chem. Theor. Comput.* **15**, 5579–5587 (2019).
- ⁴¹D. Surblys, H. Matsubara, G. Kikugawa, and T. Ohara, “Application of atomic stress to compute heat flux via molecular dynamics for systems with many-body interactions,” *Phys. Rev. E* **99**, 051301 (2019).
- ⁴²M. F. Langer, F. Knoop, C. Carbogno, M. Scheffler, and M. Rupp, “Heat flux for semi-local machine-learning potentials,” (2023), arXiv:2303.14434.
- ⁴³J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” (2018).
- ⁴⁴M. F. Langer, “glp: tools for graph machine learning potentials,” (2023).
- ⁴⁵J. E. Lennard-Jones, “On the determination of molecular fields: I. from the variation of the viscosity of a gas with temperature,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **106**, 441–462 (1924).
- ⁴⁶T. Frank, O. Unke, and K.-R. Müller, “So3krates: Equivariant attention for interactions on arbitrary length-scales in molecular systems,” in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, New Orleans, Louisiana, USA, Nov 28–Dec 9 (Curran Assoc., 2022) pp. 29400–29413.
- ⁴⁷D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature* **323**, 533–536 (1986).
- ⁴⁸N. F. Schmitz, K.-R. Müller, and S. Chmiela, “Algorithmic differentiation for automated modeling of machine learned force fields,” *J. Phys. Chem. Lett.* **13**, 10183–10189 (2022).
- ⁴⁹Non-periodic systems can be formally accommodated by setting B such that replicas lie outside the interaction cutoff radius. The glp framework supports non-periodic systems.
- ⁵⁰M. F. Langer, A. Goëßmann, and M. Rupp, “Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning,” *NPJ Comput. Mater.* **8**, 41 (2022).
- ⁵¹K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, “SchNet: A continuous-filter convolutional neural network for modeling quantum interactions,” in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Los Angeles, California, December 4–9 (Curran Assoc., 2017).
- ⁵²K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “SchNet—a deep learning architecture for molecules and materials,” *J. Chem. Phys.* **148**, 241722 (2018).
- ⁵³J. Klicpera, J. Groß, and S. Günnemann, “Directional message passing for molecular graphs,” in *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, Addis Ababa, Ethiopia, April 26–May 1 (OpenReview, 2020).
- ⁵⁴S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, “E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials,” *Nat. Commun.* **13**, 1–11 (2022).
- ⁵⁵O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, and K.-R. Müller, “SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects,” *Nat. Commun.* **12**, 7273 (2021).
- ⁵⁶I. Batatia, D. P. Kovács, G. N. C. Simm, C. Ortner, and G. Csányi, “Mace: Higher order equivariant message passing neural networks for fast and accurate force fields,” in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, New Orleans, Louisiana, USA, Nov 28–Dec 9 (Curran Assoc., 2022) pp. 11423–11436.
- ⁵⁷I. Batatia, S. Batzner, D. P. Kovács, A. Musaelian, G. N. C. Simm, R. Drautz, C. Ortner, B. Kozinsky, and G. Csányi, “The design space of E(3)-equivariant atom-centered interatomic potentials,” (2022), arXiv:2205.06643.
- ⁵⁸A. Bochkarev, Y. Lysogorskiy, C. Ortner, G. Csányi, and R. Drautz, “Multilayer atomic cluster expansion for semilocal interactions,” *Phys. Rev. Res.* **4**, L042019 (2022).
- ⁵⁹F. Knuth, C. Carbogno, V. Atalla, V. Blum, and M. Scheffler, “All-electron formalism for total energy strain derivatives and stress tensor components for numeric atom-centered orbitals,” *Comput. Phys. Comm.* **190**, 33–50 (2015).
- ⁶⁰N. C. Admal and E. B. Tadmor, “Stress and heat flux for arbitrary multibody potentials: A unified framework,” *J. Chem. Phys.* **134**, 184106 (2011).
- ⁶¹K. T. Schütt, P. Kessel, M. Gastegger, K. Nicoli, A. Tkatchenko, and K.-R. Müller, “SchNetPack: A deep learning toolbox for atomistic systems,” (2018), arXiv:1809.01072.
- ⁶²The schnetpack software library is publicly available at <https://github.com/atomistic-machine-learning/schnetpack/> under the MIT license.
- ⁶³The nequip software library is publicly available at <https://github.com/mir-group/nequip> under the MIT license.
- ⁶⁴For instance, Eq. (10) and Eq. (13) yield incorrect results if the MIC is implemented using fractional coordinates, rather than the definition using offsets.
- ⁶⁵M. S. Green, “Markoff random processes and the statistical mechanics of time-dependent phenomena,” *J. Chem. Phys.* **20**, 1281–1295 (1952).
- ⁶⁶R. Kubo, “Statistical-mechanical theory of irreversible processes. i. general theory and simple applications to magnetic and conduction problems,” *J. Phys. Soc. Jpn.* **12**, 570–586 (1957).
- ⁶⁷R. Kubo, M. Yokota, and S. Nakajima, “Statistical-mechanical theory of irreversible processes. ii. response to thermal disturbance,” *J. Phys. Soc. Jpn.* **12**, 1203–1211 (1957).
- ⁶⁸R. J. Hardy, “Energy-flux operator for a lattice,” *Phys. Rev.* **132**, 168 (1963).
- ⁶⁹A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, “The atomic simulation environment—a python library for working with atoms,” *J. Phys. Condens. Matter* **29**, 273002 (2017).
- ⁷⁰A. J. McGaughey and M. Kaviani, “Thermal conductivity decomposition and analysis using molecular dynamics simulations. part i. lennard-jones argon,” *Int. J. Heat Mass Trans.* **47**, 1783–1798 (2004).
- ⁷¹F. Knoop, T. A. R. Purcell, M. Scheffler, and C. Carbogno, “Anharmonicity in thermal insulators – an analysis from first principles,” (2022), arXiv:2209.12720.
- ⁷²“Dataset aiGK_for_anharmonic_solids on the NOMAD repository available at doi:10.17172/nomad/2021.11.11-1,” .
- ⁷³P. Vinet, J. Ferrante, J. H. Rose, and J. R. Smith, “Compressibility of solids,” *J. Geophys. Res.* **92**, 9319 (1987).
- ⁷⁴M. Hebbache and M. Zemzemi, “Ab initio study of high-pressure behavior of a low compressibility metal and a hard material: Osmium and diamond,” *Phys. Rev. B* **70**, 224107 (2004).
- ⁷⁵V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, “Ab initio molecular simulations with numeric atom-centered orbitals,” *Comput. Phys. Comm.* **180**, 2175–2196 (2009).
- ⁷⁶J. P. Perdew, A. Ruzsinszky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhou, and K. Burke, “Restoring the Density-Gradient Expansion for Exchange in Solids and Surfaces,” *Phys. Rev. Lett.* **100**, 136406 (2008).
- ⁷⁷C. Lamuta, D. Campi, L. Pagnotta, A. Dasadia, A. Cupolillo, and A. Politano, “Determination of the mechanical properties of SnSe, a novel layered semiconductor,” *J Phys Chem Solids* **116**, 306–312 (2018).
- ⁷⁸J. Brorsson, A. Hashemi, Z. Fan, E. Fransson, F. Eriksson, T. Ala-Nissila, A. V. Krashennnikov, H.-P. Komsa, and P. Erhart, “Efficient calculation of the lattice thermal conductivity by atomistic simulations with ab initio accuracy,” *Adv. Theor. Simulat.* **5**, 2100217 (2021).
- ⁷⁹H. Liu, X. Qian, H. Bao, C. Zhao, and X. Gu, “High-temperature phonon transport properties of SnSe from machine-learning interatomic potential,” *J. Phys. Condens. Matter* **33**, 405401 (2021).
- ⁸⁰P.-C. Wei, S. Bhattacharya, J. He, S. Neeleshwar, R. Podila, Y. Y. Chen, and A. M. Rao, “The intrinsic thermal conductivity of SnSe,” *Nature* **539**, E1–E2 (2016).
- ⁸¹F. Knoop, C. Carbogno, and M. Scheffler, “Ab initio Green-Kubo simulations of thermal transport in solids: Method and implementation,” (2022), arXiv:2209.01139.

Supplementary Material:

Stress and heat flux via automatic differentiation

Marcel F. Langer,^{1,2,3} J. Thorben Frank,^{1,2} and Florian Knoop⁴

¹*Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany*

²*BIFOLD – Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*

³*The NOMAD Laboratory at the Fritz Haber Institute of the Max Planck Society and Humboldt University, Berlin, Germany*

⁴*Theoretical Physics Division, Department of Physics, Chemistry and Biology (IFM), Linköping University, SE-581 83 Linköping, Sweden*

(*Electronic mail: mail@marcel.science)

(Dated: 3 May 2023)

I. HEAT FLUX WITH MINIMUM IMAGE CONVENTION

We assume that r_c^{eff} is chosen such that each atom i can interact with at most one replica of each other atom j , i.e., r_c^{eff} does not exceed half the smallest distance between opposing faces of the simulation cell. In that case, the partial derivative in Eq. 17 of the manuscript can be computed with respect to the equivalent position in the simulation cell. However, the atom-pair vector \mathbf{r}_{ji} must still connect \mathbf{r}_i in the simulation cell with the respective position \mathbf{r}_j , which may be a replica. This can be ensured by adopting the minimum image convention (MIC), yielding

$$\mathbf{J}_{\text{pot}}^{\text{MIC}} := \sum_{i,j \in \text{R}_{\text{sc}}} \left(\mathbf{r}_{ji}^{\text{MIC}} \left(\frac{\partial U_i}{\partial \mathbf{r}_j} \cdot \mathbf{v}_j \right) \right) \quad (1)$$

While this form of the heat flux requires no replica positions, it is also unsuitable for the efficient implementation with automatic differentiation (AD): As different factors are multiplied with each entry of the Jacobian $\partial U_i / \partial \mathbf{r}_j$, the evaluation of this heat flux requires the computation of the explicit full Jacobian, leading to quadratic scaling.

However, as it requires no modification in the implementation of a given potential, and can be implemented directly, albeit inefficiently, with AD, we use $\mathbf{J}_{\text{pot}}^{\text{MIC}}$ as baseline for the development of more specialized approaches.

II. LENNARD-JONES ARGON

A. Implementation

The Lennard-Jones potential used in this work is based on the ‘smooth’ implementation in ase¹, where the standard pair potential is multiplied with a cutoff function

$$f_c(r) = \begin{cases} 1 & \text{for } r < r_o \\ \frac{(r_c^2 - r^2)^2 (r_c^2 + 2r^2 - 3r_o^2)}{(r_c^2 - r_o^2)^3} & \text{for } r_o \leq r \leq r_c \\ 0 & \text{for } r > r_c, \end{cases} \quad (2)$$

ensuring that energies and forces decay continuously to zero as atoms approach r_c .

This was implemented in g1p, and energy predictions were verified to ensure that g1p and ase yield identical results.

B. Test of stress and heat flux

In order to verify the approach described in this work, and to test the g1p framework, stress and heat flux were compared with the implementation in ase, where derivatives are computed analytically. The experiment consists of computing these properties for 100 randomly perturbed simulation cells of Lennard-Jones argon, starting from an $8 \times 8 \times 8$ supercell of the face-centred cubic primitive cell with lattice parameter 3.72 \AA and angle 60° .

Positions are then modified with perturbations drawn from a normal distribution with $\sigma = 0.01 \text{ \AA}$; a random strain with each component drawn from a uniform distribution over $[-1\%, 1\%]$ is also applied. Velocities, required for the heat flux, are drawn from a Boltzmann distribution corresponding to 10 K. For the heat flux, only \mathbf{J}_{pot} is computed, as \mathbf{J}_{conv} is identical for all implementations.

III. TIN SELENIDE

A. Implementation of test and heat flux

For this experiment, the relaxed 0 K primitive cell of SnSe, obtained from the work by Knoop et al.², was extended to a $4 \times 8 \times 8$ supercell, ensuring that r_c^{eff} does not exceed half length of the smallest lattice vector. Otherwise, the experiment proceeded identically to the one described in Section II B.

Tables I and II contain comparisons of the g1p stress with finite differences for $M=1$ and $M=3$. Results are similar to the $M=2$ case.

B. Evaluating the potential

This section presents additional results with SO3KRATES and SnSe, which were used to assess the quality of the machine-learning potentials (MLPs) used in the work.

Figure 1 shows the phonon band structure for SO3KRATES with different M compared with the density-functional theory (DFT) reference. Agreement between DFT and SO3KRATES

Equation	Single		Double	
	MAE (eV)	MAPE (%)	MAE (eV)	MAPE (%)
10	$1.42 \cdot 10^{-2}$	$4.33 \cdot 10^{-2}$	$1.24 \cdot 10^{-4}$	$2.29 \cdot 10^{-4}$
11	$1.42 \cdot 10^{-2}$	$4.32 \cdot 10^{-2}$	$1.24 \cdot 10^{-4}$	$2.29 \cdot 10^{-4}$
12	$1.42 \cdot 10^{-2}$	$4.29 \cdot 10^{-2}$	$1.24 \cdot 10^{-4}$	$2.29 \cdot 10^{-4}$
13	$1.42 \cdot 10^{-2}$	$4.33 \cdot 10^{-2}$	$1.24 \cdot 10^{-4}$	$2.29 \cdot 10^{-4}$
14	$1.42 \cdot 10^{-2}$	$4.32 \cdot 10^{-2}$	$1.24 \cdot 10^{-4}$	$2.29 \cdot 10^{-4}$
15	$1.42 \cdot 10^{-2}$	$4.29 \cdot 10^{-2}$	$1.24 \cdot 10^{-4}$	$2.29 \cdot 10^{-4}$

TABLE I. Error in stress for tin selenide, comparing different formulations to finite differences, for SO3KRATES with $M=1$. Results are shown for both single and double precision arithmetic, and for $\sigma \cdot V$ in place of σ .

Equation	Single		Double	
	MAE (eV)	MAPE (%)	MAE (eV)	MAPE (%)
10	$2.07 \cdot 10^{-2}$	$4.86 \cdot 10^{-2}$	$1.91 \cdot 10^{-4}$	$3.71 \cdot 10^{-4}$
11	$2.07 \cdot 10^{-2}$	$4.87 \cdot 10^{-2}$	$1.91 \cdot 10^{-4}$	$3.71 \cdot 10^{-4}$
12	$2.06 \cdot 10^{-2}$	$4.86 \cdot 10^{-2}$	$1.91 \cdot 10^{-4}$	$3.71 \cdot 10^{-4}$
13	$2.07 \cdot 10^{-2}$	$4.86 \cdot 10^{-2}$	$1.91 \cdot 10^{-4}$	$3.71 \cdot 10^{-4}$
14	$2.07 \cdot 10^{-2}$	$4.87 \cdot 10^{-2}$	$1.91 \cdot 10^{-4}$	$3.71 \cdot 10^{-4}$
15	$2.06 \cdot 10^{-2}$	$4.86 \cdot 10^{-2}$	$1.91 \cdot 10^{-4}$	$3.71 \cdot 10^{-4}$

TABLE II. Error in stress for tin selenide, comparing different formulations to finite differences, for SO3KRATES with $M=3$. Results are shown for both single and double precision arithmetic, and for $\sigma \cdot V$ in place of σ .

increases systematically with M ; satisfactory agreement is obtained from $M=2$ onwards.

The vibrational density of states can be seen in Fig. 2. Overall features are captured for all values of M . Performance is slightly improved for $M=2, 3$, which behave similarly.

In line with observations in the main text, $M=2$ can be considered sufficient to model vibrational properties. While slight improvements are obtained with $M=3$, the impact on thermal conductivity is negligible. Energy-volume curves, on the other hand, are improved by $M=3$.

C. Green-Kubo workflow and convergence

As the Green-Kubo (GK) method is formally valid in the thermodynamic limit, the convergence of κ with respect to simulation size and duration must be considered. Additional considerations arise from the use for noise reduction in the resulting heat flux autocorrelation function (HFACF). In this work, following previous work^{3,4} a lowpass filter with frequency 1 THz is applied to the integrated HFACF, which is then differentiated via finite differences to yield a smooth HFACF from which the integration cutoff can be determined as the first zero crossing. No further noise reduction is performed; no gauge terms are removed and the full heat flux $\mathbf{J} = \mathbf{J}_{\text{pot}} + \mathbf{J}_{\text{conv}}$ is computed at all timesteps.

Eleven trajectories with timestep $\Delta t = 4$ fs are used throughout, initialised from snapshots from a 0.2 ns NVT thermalization trajectory using the Langevin thermostat implemented

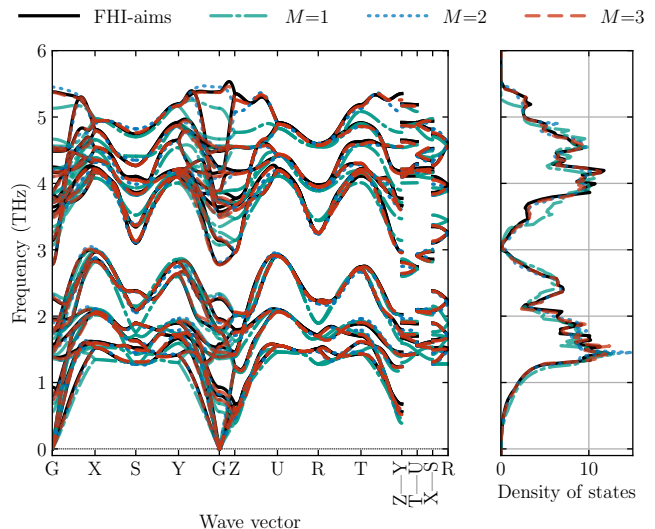


FIG. 1. Phonon band structure and density of states for tin selenide for SO3KRATES models with $M=1, 2, 3$ compared with FHI-aims. Results are reported for a supercell with 256 atoms.

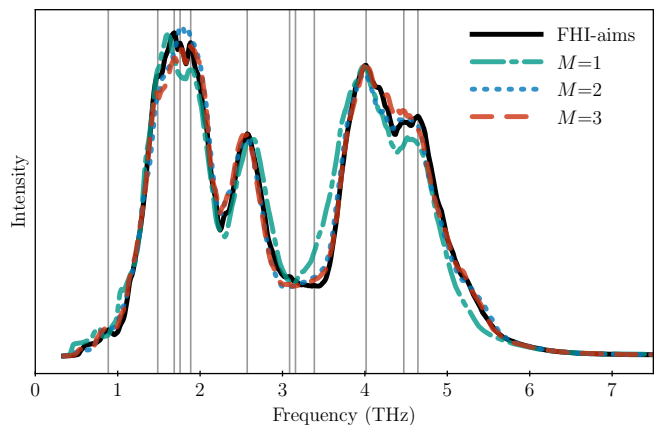


FIG. 2. Vibrational density of states for tin selenide at 300 K for SO3KRATES models with $M=1, 2, 3$ compared with FHI-aims. Results are reported for a trajectory with 30 ps duration in a supercell with 256 atoms, started from identical initial configurations. Vertical lines indicate prominent peaks in the FHI-aims result.

in ase¹. The primitive cell at 300 K from the work by Knoop et al.² was used, creating $n \times 2n \times 2n$ supercells to obtain simulation cells at different $N = 32n^3$.

Figure 3 displays the convergence of κ with respect to simulation cell size N and simulation duration t_0 , for SO3KRATES with $M=2$. No significant increases in κ are observed for simulation cell sizes above $N=864$ and simulation durations above $t_0=2$ ns. We therefore choose (864, 2 ns) as ‘production’ settings for this work.

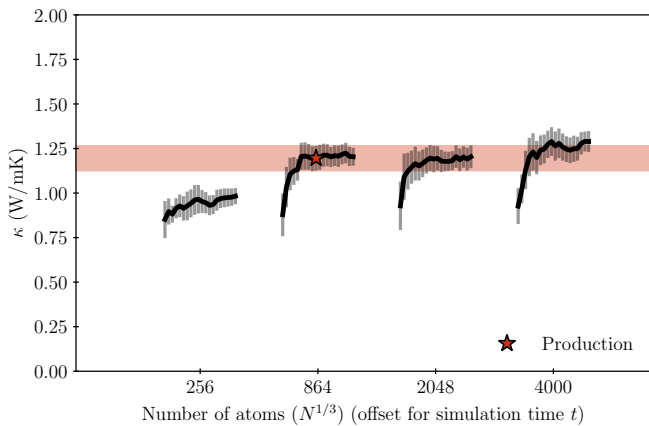


FIG. 3. κ for tin selenide at 300 K for different choices of simulation cell size N and simulation duration t_0 for SO3KRATES with $M=2$. Error bars indicate the standard error across trajectories. N is shown as $N^{1/3}$, which is proportional to the length scale of the simulation cell. For each choice of N , t_0 from 0.1 ns to 4.0 ns are shown with a horizontal offset. ‘Production’ settings (864, 2 ns) are indicated; the associated standard error is shown as a shaded band.

D. Model and training

Here we use the SO3KRATES message-passing neural network (MPNN), which is implemented in the mlff package⁵. The number of interaction steps M is varied from $M = 1$ up to $M = 3$, whereas we fix the cutoff radius to $r_c = 5 \text{ \AA}$, the embedding dimension to $F = 132$ and the maximal degree in the equivariant branch to $l_{\max} = 3$. Non-local corrections are not used. After M message-passing updates the final embeddings are put through a two layered feed-forward neural network (FFNN) with SILU (sigmoid linear unit) non-linearity to obtain per-atom energies. The total potential energy is given as the sum of the per-atom energies.

The model is trained by minimizing a joint loss of the potential energy and per-atom forces with loss weightings of 0.01 and 0.99, respectively using the ADAM optimizer⁶. In total 3000 reference structures of tin selenide are used for training, of which 600 are reserved for validation. The training is stopped after 2500 epochs, with a batch size of 10. After each epoch, the performance of the current model is evaluated on the validation data and the best-performing model is saved for production. The initial learning is set to 10^{-3} and is reduced every 100k steps using exponential learning rate decay with a decay factor of 0.7. No early stopping is employed. Training times on a single NVIDIA A100 40GB GPU range from 2h54min for $M = 1$ up to 6h46min for $M = 3$.

IV. EQUATION OF STATE (EOS)

To describe cohesive properties, we use the Vinet equation of state^{7–10} for pressure p as function of volume V :

$$p(V) = \frac{3B_0}{X^2} (1-X)e^{\eta(1-X)}, \quad (3)$$

with

$$X = \left[\frac{V}{V_0} \right]^{\frac{1}{3}} \quad \text{and} \quad \eta = \frac{3}{2} (B'_0 - 1), \quad (4)$$

where V_0 is the volume at vanishing pressure, B_0 the bulk modulus and B'_0 its volume derivative. The respective energy function that fulfills $p = -dE/dV$ is given by¹¹

$$E(X) = E_0 + \frac{2B_0V_0}{(B'_0 - 1)^2} \left(2 - (5 + 3B'_0(X - 1) - 3X) e^{\eta(1-X)} \right). \quad (5)$$

These functions are fitted to the computed values for E or p and V to obtain V_0 , B_0 , B'_0 , and the minimal energy E_0 in the case of Eq. (5).

¹A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, “The atomic simulation environment—a python library for working with atoms,” *J. Phys. Condens. Matter* **29**, 273002 (2017).

²F. Knoop, T. A. R. Purcell, M. Scheffler, and C. Carbogno, “Anharmonicity in thermal insulators – an analysis from first principles,” (2022), arXiv:2209.12720.

³F. Knoop, C. Carbogno, and M. Scheffler, “Ab initio Green-Kubo simulations of thermal transport in solids: Method and implementation,” (2022), arXiv:2209.01139.

⁴M. F. Langer, F. Knoop, C. Carbogno, M. Scheffler, and M. Rupp, “Heat flux for semi-local machine-learning potentials,” (2023), arXiv:2303.14434.

⁵The mlff software library is publicly available at <https://github.com/thorben-frank/mlff/>.

⁶D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” (2014), arXiv:1412.6980.

⁷J. H. Rose, J. R. Smith, F. Guinea, and J. Ferrante, “Universal features of the equation of state of metals,” *Phys. Rev. B* **29**, 2963–2969 (1984).

⁸P. Vinet, J. Ferrante, J. R. Smith, and J. H. Rose, “A universal equation of state for solids,” *J Phys C Solid State Phys* **19**, L467 (1986).

⁹P. Vinet, J. R. Smith, J. Ferrante, and J. H. Rose, “Temperature effects on the universal equation of state of solids,” *Phys. Rev. B* **35**, 1945–1953 (1987).

¹⁰P. Vinet, J. Ferrante, J. H. Rose, and J. R. Smith, “Compressibility of solids,” *J. Geophys. Res.* **92**, 9319 (1987).

¹¹M. Hebbache and M. Zemzemi, “Ab initio study of high-pressure behavior of a low compressibility metal and a hard material: Osmium and diamond,” *Phys. Rev. B* **70**, 224107 (2004).