Shift-Collapse Acceleration of Generalized Polarizable Reactive Molecular Dynamics for Machine Learning-Assisted Computational Synthesis of Layered Materials

Kuang Liu†, Subodh Tiwari†, Chunyang Sheng†, Aravind Krishnamoorthy†, Sungwook Hong†, Pankaj Rajak†, Rajiv K. Kalia†, Aiichiro Nakano†, Ken-ichi Nomura†, Priya Vashishta†, Manaschai Kunaseth‡, Saber Naserifar§, William A. Goddard III§, Ye Luo*, Nichols A. Romero*, Fuyuki Shimojo¶

† Collaboratory for Advanced Computing and Simulations, University of Southern California
‡ National Science and Technology Development Agency, Thailand
§ Materials and Process Simulation Center, California Institute of Technology
* Argonne Leadership Computing Facility, Argonne National Laboratory
¶ Department of Physics, Kumamoto University, Japan
†{liukuang, sctiwari, chunyangs, kris658, sungwooh, rajak, rkalia, anakano, knomura, priyav}@usc.edu, ‡manaschai@nanotec.or.th, §naseri@caltech.edu, §wag@wag.caltech.edu, *{yeluo, naromero}@anl.gov, ¶shimojo@kumamoto-u.ac.jp

***Abstract* — Reactive molecular dynamics is a powerful simulation method for describing chemical reactions. Here, we introduce a new generalized polarizable reactive force-field (ReaxPQ+) model to significantly improve the accuracy by accommodating the reorganization of surrounding media. The increased computation is accelerated by (1) extended Lagrangian approach to eliminate the speed-limiting charge iteration, (2) shift-collapse computation of many-body renormalized *n*-tuples, which provably minimizes data transfer, (3) multithreading with round-robin data privatization, and (4) data reordering to reduce computation and allow vectorization. The new code achieves (1) weak-scaling parallel efficiency of 0.989 for 131,072 cores, and (2) eight-fold reduction of time-to-solution (T2S) compared with the original code, on an Intel Knights Landing-based computer. The reduced T2S has for the first time allowed purely computational synthesis of atomically-thin transition metal dichalcogenide layers assisted by machine learning to discover a novel synthetic pathway.**

Keywords — Applications/ Computational materials science and engineering; Algorithms/Hybrid/heterogeneous/accelerated algorithms and other high-performance algorithms.

I. Introduction: Importance of the Problem

Reactive molecular dynamics (RMD) is a powerful simulation method for describing material processes involving chemical reactions, with a wide range of applications in physics, chemistry, biology and materials science [1]. RMD simulation follows time evolution of the positions, $\mathbf{r}^N = \{\mathbf{r}_i | i = 1, \ldots, N\}$, of N atoms by numerically integrating Newton's equations of motion, where the atomic force law is mathematically encoded in the interatomic potential energy $E(\mathbf{r}^N)$. Reliable interatomic potentials are key to accurately describing thermomechanical and chemical properties of materials. The first principles-informed reactive force-field (ReaxFF) model significantly reduces the computational cost, while reproducing the energy surfaces and barriers as well as charge distributions of quantum-mechanical (QM) calculations [1].

The most intensive computation in RMD simulation arises from a charge-equilibration (QEq) scheme [2] to describe charge transfer between atoms, thereby enabling the study of reduction and oxidation reactions. QEq treats atomic charges as dynamic variables, $q^N = \{q_i | i = 1, \ldots, N\}$. The charges and the resulting force law are determined by minimizing the potential energy with respect to $\boldsymbol{q}^N$ at every RMD time step. This variable N-charge problem is commonly solved iteratively, *e.g.*, with the conjugate gradient (CG) method [3]. Though recent advancements in parallel ReaxFF algorithms [4-6] have enabled large RMD simulations [7] involving multimillion atoms, QEq computation remains to be the major bottleneck for studying long time trajectories of such large RMD simulations.

Despite enormous success of the QEq-based ReaxFF model, one critical issue has remained unsolved, namely accurate description of electric polarizability. Polarization of the surrounding medium essentially dictates the rate of reduction and oxidation reactions, as is articulated, *e.g.*, in the Nobel lecture by Rudolph Marcus [8]. A recently proposed polarizable reactive force-field (ReaxPQ) model based on a polarizable charge equilibration (PQEq) scheme significantly improves the accuracy of describing redox reactions by accommodating the reorganization of surrounding media [9]. When applied to prediction of electronic polarizabilities, however, the ReaxPQ model alone was found inadequate. This partly arises from the fact that the original ReaxPQ model determines the polarization of a nucleus core and an electronic shell within each atom by considering only the internal electric field produced by atomic charges but not an externally applied electric field. To remedy this deficiency, we here introduce a generalization of the ReaxPQ model named ReaxPQ+, in which atomic polarizations respond to both internal and external electric fields, thereby achieving near quantum accuracy for tested cases.

The improved accuracy of the new ReaxPQ+ model is accompanied by significant increase of the computational cost.

978-1-7281-0176-7/18/$31.00 ©2018 IEEE
DOI 10.1109/ScalA.2018.00009

Compared to the original QEq scheme, which only deals with atom-centered charge-charge interactions, PQEq computation in the ReaxPQ+ model is quadrupled, since it considers core-core, core-shell, shell-core and shell-shell charge interactions for every atomic pair. In this paper, we accelerate this heavy ReaxPQ+ computation using (1) an extended Lagrangian approach to eliminate the speed-limiting charge iteration [10], (2) a new extension of the shift-collapse (SC) algorithm [11] named renormalized SC (RSC) to compute dynamic n-tuples with provably minimal data transfers, (3) multithreading with round-robin data privatization, and (4) data reordering to reduce computation and allow vectorization. The accelerated code achieves (1) weak-scaling parallel efficiency of 0.989 for 131,072 cores, and (2) eight-fold reduction of the time-to-solution (T2S) compared with the original code, on an Intel Knights Landing (KNL)-based supercomputer.

The reduced T2S has allowed computational synthesis of atomically-thin transition-metal dichalcogenide (TMDC) layers with unprecedented fidelity. Functional layered materials (LM) will dominate materials science in this century [12]. The attractiveness of LMs lies not only in their outstanding electronic, optical, magnetic and chemical properties, but also in the possibility of tuning these properties in desired ways by building van der Waals (vdW) heterostructures composed of unlimited combinations of atomically-thin layers. To rationally guide the synthesis of stacked LMs by chemical vapor deposition (CVD), exfoliation and intercalation, "computational synthesis" should encompass large spatiotemporal scales. Such layered materials genome (LMG) has been chosen as one of the designated applications of the United States' first exaflop/s computer A21 when it is introduced in 2021 [13]. This paper for the first time demonstrates purely computational synthesis of TMDC-based LM, which is assisted by machine learning to discover a novel synthetic pathway. This opens up a possibility to computationally explore new synthetic pathways to novel TMDC-LMs and vdW heterostructures.

II. Application and Algorithmic Innovations

A. Generalized Polarizable Reactive Force Field (ReaxPQ+)

In the ReaxPQ+ model, the potential energy $E(\{\mathbf{r}_{ij}\},\{\mathbf{r}_{ijk}\},\{\mathbf{r}_{ijkl}\},\{q_i\},\{BO_{ij}\})$ is a function of relative positions of atomic pairs $\mathbf{r}_{ij}$, triplets $\mathbf{r}_{ijk}$ and quadruplets $\mathbf{r}_{ijkl}$, as well as atomic charges q_i and bond orders BO_{ij} between atomic pairs. The potential energy includes Coulombic energy E_{Coulomb}. In the PQEq scheme used in the original ReaxPQ model, electric polarization is described using a Gaussian-shaped electron density (shell) that can polarize away from the nuclei core in response to internal electric fields produced by atoms [9]. Here, each atom i is partitioned into two charged sites (*i.e.*, core and shell). The core (ρ_{ic}) consists of two parts: ρ_i with a variable total charge (q_i) and ρ_{iZ} with a fixed total charge (Z_i). The shell (ρ_{is}) is massless and has a fixed total charge of $-Z_i$. The shell and core of an atom are connected by an isotropic harmonic spring with force constant K_s (Fig. 1).

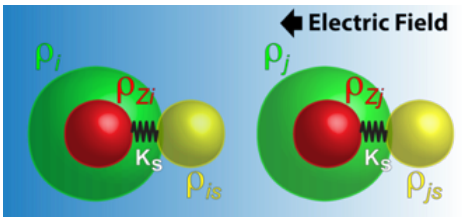

Fig. 1. Schematic of the response of core (green) and shell (yellow) charges to an external electric field in the new PQEq+ model.

At each time step, the atomic charges q^N are determined by minimizing E_{Coulomb} subject to the conditions that the electrochemical potentials, $\partial E_{\text{Coulomb}}/\partial q_i$, are equal among all atoms. The Coulombic energy is given by

$$\begin{aligned} E_{\text{Coulomb}}(\{\mathbf{r}_{ic},\mathbf{r}_{is},q_i\}) = \sum_{i=1}^{N}\left[\chi_i^0 q_i + \frac{1}{2}J_{ii}^0 q_i^2 + \frac{1}{2}K_s r_{ic,is}^2\right] \\ + \sum_{i>j}[T(r_{ic,jc})C_{ic,jc}(r_{ic,jc})q_{ic}q_{jc} \\ - T(r_{ic,js})C_{ic,js}(r_{ic,js})q_{ic}Z_j \\ - T(r_{is,jc})C_{is,jc}(r_{is,jc})q_{jc}Z_i \\ + T(r_{is,js})C_{is,js}(r_{is,js})Z_iZ_j] \end{aligned} \quad (1)$$

where $\mathbf{r}_{ic}$, $\mathbf{r}_{is}$, χ_i^0 and J_{ii}^0 are the core position, shell position, electronegativity and hardness of the i-th atom. In Eq. (1), $r_{ia,jb}$ ($i, j = 1,\ldots, N$; a, b = core(c) or shell(s)) are charge-charge distances. The electrostatic energy between two Gaussian charges is given in terms of the error function $C_{ia,jb}(r_{ia,jb})$, and the Coulombic interaction is screened using a taper function $T(r)$ [9]. As was mentioned in the introduction, the core-core, core-shell, shell-core and shell-shell charge interactions in Eq. (1) quadruple the charge computation over the conventional ReaxFF model.

In the original ReaxPQ, the shell position $\mathbf{r}_{is}$ for the i-th atom is obtained by balancing the effect of the electrostatic field due to all other atoms (*i.e.*, inter-atomic interactions) with intra-atomic interactions involving only the core and shell:

$$\mathbf{F}_{\text{inter}} = -\frac{\partial}{\partial \mathbf{r}_{is}}\{\textstyle\sum_{ia>jb} T(r_{ia,jb})C_{ia,jb}(r_{ia,jb})q_{ia}q_{jb}\} \quad (2)$$

$$\mathbf{F}_{\text{intra}} = -\frac{\partial}{\partial \mathbf{r}_{is}}\left(\frac{1}{2}K_s r_{ic,is}^2\right) \quad (3)$$

We solve $\mathbf{F}_{\text{inter}} = \mathbf{F}_{\text{intra}}$ to determine $\mathbf{r}_{is}$ using Newton-Raphson method. Fig. 2(a) compares time evolution of the PQEq energy for 1, 10 and 100 iterations of Newton-Raphson method, where the previous shell positions are used as initial guess. Fig. 2(a) shows that single iteration suffices to obtain the accuracy of 10^{-4} kcal•mol^{-1}/atom. We have also confirmed the accuracy of solution by comparing the shell position and charge of each atom. We have applied this ReaxPQ model to compute the dielectric constants ϵ of various materials and found that the model generally underestimates the values.

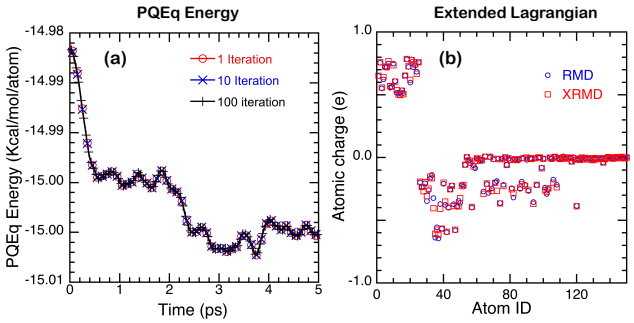

Fig. 2. (a) PQEq energy with 1, 10 and 100 iterations of Newton-Raphson method. (b) Atomic charges after 1 ps of MD simulation with converged RMD (blue) and XRMD (red) methods.

In order to improve the accuracy of describing the dielectric response of materials, we here introduce a generalized polarizable reactive force-field (ReaxPQ+) model, in which $\mathbf{r}_{is}$

is determined by solving $\mathbf{F}_{\text{inter}} + \mathbf{F}_{\text{external}} = \mathbf{F}_{\text{intra}}$, *i.e.*, by explicitly including the effect of an external electric field $\boldsymbol{\mathcal{E}}$,

$$\mathbf{F}_{\text{external}} = \sum_a q_{ia}\boldsymbol{\mathcal{E}} \quad (4)$$

In both the original ReaxPQ and new ReaxPQ+ models, polarization is calculated as

$$\mathbf{P} = \sum_{ia} q_{ia}(\mathbf{r}_{ia} - \mathbf{r}_{ia}^0) \quad (5)$$

where $\mathbf{r}_{ia}^0$ is the charge position in the absence of external electric field $\boldsymbol{\mathcal{E}}$.

TABLE I. Computed dielectric constants.

Material	ReaxPQ	ReaxPQ+	QM
Polyethylene (PE)	1.01	2.25	2.37
C=O defect (PE)	1.02	2.74	2.78
C-Cl defect (PE)	1.01	2.40	2.53
MoS_2 (in plane)	1.03	14.3	15.4*
MoS_2 (out of plane)	1.03	5.68	7.43*
Polyvinylidene fluoride (PVDF)	1.02	2.56	2.52
Alumina	1.03	3.17	2.98

**Surface Science Reports,* vol. 70, pp. 554-586, Dec 2015.

We have tested the accuracy of the ReaxPQ+ model by computing the dielectric constants ϵ of poly-ethylene (PE), molybdenum disulfide (MoS_2) and other polymer and ceramic materials. Table I compares dielectric constants computed with the original ReaxPQ and new ReaxPQ+ models against those obtained by first-principles QM calculations. As noted otherwise, the QM value has been computed by us using a quantum molecular dynamics (QMD) code [14] based on density functional theory (DFT), in which dielectric constants are calculated using a Berry-phase approach [15]. The table shows that the new ReaxPQ+ results agree much better with the QM results, compared with the original ReaxPQ results. The improved accuracy has been confirmed for broad organic and inorganic materials with varying moiety, anisotropy and defects, which will be published elsewhere.

B. Extended-Lagrangian Acceleration of ReaxPQ+

As shown above, charge-interaction computation in ReaxPQ+ is quadrupled compared to that in the conventional ReaxFF model. The increased computational cost necessitates innovative algorithms to speed up the computation.

First, we adapt the extended-Lagrangian reactive molecular dynamics (XRMD) algorithm [10], which was originally proposed for the conventional ReaxFF model, to the new ReaxPQ+ model. The problem is that an excessively large number of CG iterations are required to reach sufficient convergence of charges q^N to guarantee the conservation of the total energy as a function of time. Insufficiently converged charges act as an artificial heat sink, and the resulting broken time reversibility causes the total energy to drift over time. A similar trade-off between the computational speed and energy conservation is encountered in first-principles QMD simulations, where insufficient convergence of the iterative refinement of electronic wave functions causes serious energy drift. Niklasson proposed an extended Lagrangian scheme [16] that achieves excellent long-time energy conservation with drastically reduced number of iterations. In fact, an extended Lagrangian scheme with no iteration (*i.e.*, requiring only one evaluation of energy gradient per QMD time step) has been demonstrated [17]. The key idea is to introduce auxiliary wave functions as dynamic variables that are numerically integrated by time-reversible, symplectic integration schemes to address the broken reversibility problem, while the auxiliary wave functions are constrained to iteratively determined wave functions by a harmonic potential. Successful elimination of the speed-limiting charge iteration in the ReaxFF model was achieved by Nomura *et al*. by adapting the extended-Lagrangian scheme [10]. The XRMD algorithm has drastically improved energy conservation while substantially reducing the time-to-solution. In addition, XRMD accurately describes atomic trajectories and charges. The average difference of atomic positions was 0.08 Å after 1 ps of simulation between XRMD and fully converged RMD methods [10]. Fig. 2(b) compares atomic charges obtained by XRMD algorithm with those by RMD using extremely high CG tolerance (10^{-8}) which show an excellent agreement. In this paper, we adapt the XRMD algorithm to the new ReaxPQ+ model, where auxiliary charges are applied only to the variable part, q_i, of the charges.

C. Renormalized Shift-Collapse Acceleration of ReaxPQ+

Our second algorithmic innovation is a generalization of the shift-collapse (SC) algorithm [11], named renormalized SC (RSC). SC algorithm provably minimizes data transfer for computation of dynamic *n*-tuples in parallel computing based on spatial (or domain) decomposition. Building on translation and reflection invariance of the set of *n*-tuple computations, the algorithm applies shift and collapse algebraic transformations to *n*-tuple computations so as to completely eliminate redundant computations among computing nodes while minimizing data transfer. Here, we apply the SC algorithm to the generalized polarizable charge equilibration (PQEq+) subroutine that iteratively optimizes atomic charges to minimize the Coulombic potential energy using CG method. At the beginning of every PQEq+ iteration, information of neighbor atoms near the domain boundary (including atomic charges and gradients) needs to be imported from the nearest-neighbor computing nodes for calculating Coulombic potential used in CG iteration. This incurs significant communication cost. For each PQEq+ iteration, Coulombic potential of the system is evaluated using guessed atomic charges. After that, each node computes charge gradients and consecutively updates atomic charges of all atoms resided in its domain. PQEq+ computation is terminated when either (1) CG residue is small enough, (2) no energy improvement is gained between successive iterations, or (3) maximum number of iterations is reached.

In this work, we employ two approaches to reduce the time spent in PQEq+ subroutine. First, we apply SC algorithm to minimize communication cost and eliminate redundant pair evaluations within PQEq+ subroutine. Here, we also develop a new SC approach to handle many-body renormalized *n*-tuple computation in PQEq+ due to the interaction with surrounding atoms, which has never been addressed by SC algorithm before. Second, we store-and-reuse unchanged coefficients across multiple CG iterations, thereby improving the efficiency of SC computation. Details of these approaches are discussed below.

SC algorithm utilizes 3-step communication, thereby significantly reducing both bandwidth and latency costs when compared to the 6-step communication (*i.e.* halo exchange) in conventional full-shell (FS) method [18]. Computation pattern of SC algorithm also eliminates redundant pair evaluations. Evaluating Coulombic potential using SC algorithm is rather straightforward. The computation follows SC computation

pattern for two-body interaction, which includes pairs of resident atoms and pairs of atoms in the neutral territory (NT) [19]. Typically, contribution from pairs in NT region needs to be sent back to its resident node. However, the return of contribution is not required in this situation. Here, only overall sum of Coulombic potential contributions from every node is needed, which can be efficiently achieved by using the MPI_AllReduce function in the message passing interface (MPI) standard. Nevertheless, subsequent computation after obtaining total Coulombic potential requires special treatment for many-body computation within the CG subroutine. Namely, computation of charge gradient $g(i)$ for a particular atom i requires sum of contributions over all neighbor atoms of i:

$$g(i) = \sum_{j\epsilon \mathrm{NBR}(i)} H_{ij} q_j \quad (6)$$

where H_{ij} denotes charge Hessian of atom pair i and j, and NBR(i) is the neighbor list of atoms i. Computing $g(i)$ involves many-body renormalization because it requires H_{ij} contributions from all neighbor atoms in NBR(i); see Fig. 3(a). However, in SC computation, atoms near the lower domain boundary may not have complete neighbor-atom information in the node it resides. In fact, neighbor-atom information is completed when combining partial neighbor lists (PNBR) from multiple computing nodes. As such, gradient calculation in SC algorithmic framework $g_{\mathrm{SC}}(i)$ is defined as

$$g_{\mathrm{SC}}(i) = \sum_{j\epsilon \mathrm{PNBR}^{\mathrm{RES}}(i)} H_{ij} q_j + \sum_k \left(\sum_{j'\epsilon \mathrm{PNBR}_k^{\mathrm{NT}}(i)} H_{ij'} q_{j'} \right) \quad (7)$$

where $\mathrm{PNBR}^{\mathrm{RES}}(i)$ denotes partial neighbor list of atom i in the node that atom i resides. $\mathrm{PNBR}_k^{\mathrm{NT}}(i)$ denotes partial neighbor list of atom i in NT region from node k, which is not in the same node that i resides. Therefore, to complete $g(i)$ calculation in SC framework, partial $g(i)$ contribution based on $\mathrm{PNBR}^{\mathrm{NT}}(i)$ must be sent back to the node that owns i (resident node); see Fig. 3(b). Although this incurs extra communication (additional 3-way communication for returning $g_{\mathrm{SC}}(i)$ contribution), this is still no larger than the 6-way communication in conventional FS method. Furthermore, FS computation pattern yields substantial computational overhead from redundant pair evaluations to build complete NBR in every computing node. On the other hand, SC algorithm performs only essential pair evaluations without redundancy. This is expected to significantly reduce running time inside PQEq+ subroutine, while maintaining the same communication cost.

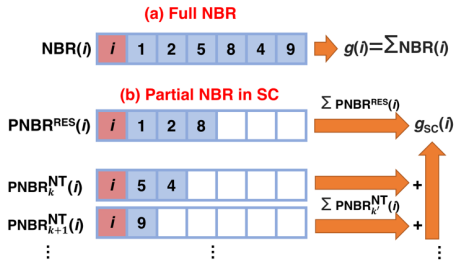

Fig. 3. Illustration of many-body renormalized n-tuple computation of $g(i)$ based on (a) conventional full neighbor list (NBR) and (b) partial neighbor lists ($\mathrm{PNBR}^{\mathrm{RES}}$ and $\mathrm{PNBR}^{\mathrm{NT}}$) used in the RSC algorithm.

For each PQEq+ iteration, Coulombic potential energy between atoms i and j based on (1) takes the following form:

$$E_{\mathrm{Coulomb}}(i,j) = E_{\text{core-core}}(i,j) + E_{\text{core-shell}}(i,j) + E_{\text{core-shell}}(j,i) + E_{\text{shell-shell}}(i,j)$$

$$E_{\text{core-core}}(i,j) = T(r_{ic,jc}) C_{ic,jc}(r_{ic,jc}) q_i q_j$$

$$E_{\text{core-shell}}(i,j) = -T(r_{ic,js}) C_{ic,js}(r_{ic,js}) q_i Z_j \quad (8)$$

$$E_{\text{shell-shell}}(i,j) = T(r_{is,js}) C_{is,js}(r_{is,js}) Z_i Z_j$$

Here, $T(r_{ia,jb})$ and $C_{\mathrm{ia,jb}}(r_{ia,jb})$ are computed using a costly table lookup as a function of core/shell distance. However, only q_i is changing across PQEq+ iterations, while $T(r_{ia,jb})$ and $C_{\mathrm{ia,jb}}(r_{ia,jb})$ remain unchanged (*i.e.* atomic/shell positions are fixed). Therefore, we save considerable computation time by storing these four unchanged coefficients $T(r_{ia,jb}) C_{ia,jb}(r_{ia,jb})$ for each atomic pair throughout PQEq+ iterations.

III. Parallel Implementation and Performance Optimizations

We have implemented RMD simulation based on the new algorithmically-accelerated ReaxPQ+ model in section II in a scalable parallel RMD code named RXMD [4]. In this code, computations are parallelized using spatial decomposition, where the simulated system is decomposed into spatially localized subsystems and each processor is assigned computations associated with one subsystem. Message passing is used to exchange necessary data for the computations between processors, utilizing the MPI standard. Specifically, before computing the forces on atoms in a subsystem, atomic positions within the interaction cutoff radius within the boundaries of the neighboring subsystems are copied from the corresponding processors (*i.e.*, inter-process atom caching). After updating the atomic positions according to time-stepping, some atoms may have moved out of its subsystem. These moved-out atoms are migrated to the proper neighbor processors (*i.e.*, inter-process atom migration). The RXMD code is written in Fortran 90.

For large granularity (the number of atoms per spatial subsystem, $N/D > 10^2$), spatial decomposition (*i.e.*, each processor is responsible for the computation of the forces on the atoms within its subsystem) suffices. For finer granularity ($N/D \sim 1$), on the other hand, neutral-territory (NT) [19] or other hybrid decomposition schemes is more efficient. As discussed in section II, we use the SC scheme [11], which is a generalization of NT for general dynamic n-tuple computation.

Grain size for each MPI rank is limited by the cutoff length of interaction. To further accelerate the computation within MPI rank, we introduce an additional layer of shared-memory parallelism using the Open Multi-Processing (OpenMP) application programming interface. This hierarchical MPI+OpenMP implementation allows RXMD to take advantage of the simultaneous multithreading support provided by modern processors to achieve better utilization of the computing resources within each processor. With multithreading, the most computationally expensive bond-order and force computations within RXMD are greatly accelerated, serving to reduce the overall runtime. A secondary benefit of multithreading is that it allows MPI ranks to be exchanged for local threads, thereby reducing the total number of ranks in a MPI job and similarly reducing the communication and atom-caching overheads at large scales. To obtain the best time-to-solution (T2S) on each Intel Knights Landing (KNL) node, we performed timed run for several possible configurations of MPI ranks and OpenMP threads, as shown in Fig. 4(a). In this test, the product of the number of MPI ranks and that of OpenMP threads is fixed to 64. The best T2S was observed for the

combination of 16 MPI ranks and 4 OpenMP threads on each node, which will be used in subsequent benchmark tests.

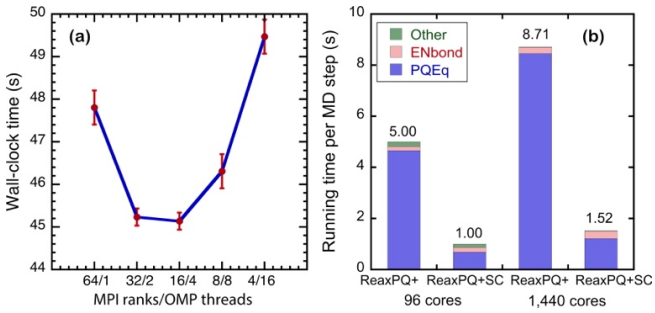

Fig. 4. (a) Runtime comparison between different configurations of MPI ranks and OpenMP threads on 1 node to identify the optimal combination. Total number of processes is 64. (b) Average runtime per step for the original ReaxPQ+ and new ReaxPQ+SC implementations on 96 and 1,440 cores. ReaxPQ+SC reduces time-to-solution 5.0- and 5.7-fold below ReaxPQ+ for 96- and 1,440-core benchmarks, respectively. The numbers denote runtimes.

A. Performance Improvement by Algorithmic Accelerations

We first test the effect of SC acceleration described in section II.C. Fig. 4(b) shows the average runtime per time step on 96 cores (left) and 1,440 cores (right). In each case, the left and right bars show the total runtime without (ReaxPQ+) and with (ReaxPQ+SC) SC acceleration, respectively. The figure also shows partial runtimes for polarizable charge-equilibration (PQEq) calculation, which has become the speed-limiting step in the new ReaxPQ+ model in order to achieve the improved accuracy demonstrated in section II.A, as well as for non-bonded (ENbond) and other force calculations. We observe that the SC-accelerated ReaxPQ+SC is 5.0 and 5.7 times faster than the original implementation, ReaxPQ+, on 96 and 1,440 cores, respectively (N/P = 672). We should note that this significant speedup is a result of the following two improvements. First, the SC framework minimized the computation by removing redundant pair calculations, which is especially beneficial for fine granularity (*i.e.*, small number of atoms per computing node). The reduced communication in ReaxPQ+SC is highlighted by the increased relative performance with ReaxPQ+ from 5.0 to 5.7 fold when the number of cores increases from 96 to 1,440. Second, the computation was further reduced by reusing Coulombic coefficients.

B. Improving Thread Scalability by Round-Robin Data Privatization

In molecular dynamics (MD) simulations like RMD, interatomic forces need to be updated at every time step, which consumes substantial computing. With shared-memory programming, force calculation is parallelized by distributing computations to threads. Because of Newton's third law of motion, it is possible that different threads compute pair interactions involving a common atom and update the force on that atom simultaneously, thereby causing memory conflict. To avoid such race conditions, OpenMP provides atomic directive to handle concurrent memory access with a few locks.

In the RXMD code, interatomic forces for all atoms within an MPI process are stacked in an array f of length n, where n is the total number of atoms per MPI rank. Array f is shared and atomically accessed by threads in force-calculation module (Fig. 5(a)). However, overhead of concurrent memory access to array f can become performance bottleneck as the simulation size grows. We observed that the cumulative CPU time of atomic operations accounts for 40% of the total runtime. Fig. 5(b) shows how a thread-privatization scheme addresses this issue [20]. Specifically, the program allocates a private copy, *f_private*, of array f for each thread and accesses *f_private* individually. When a piece of calculated force data on a common atom needs to be updated by more than one threads, these threads write partial results to their own arrays concurrently, thereby eliminating atomic operations.

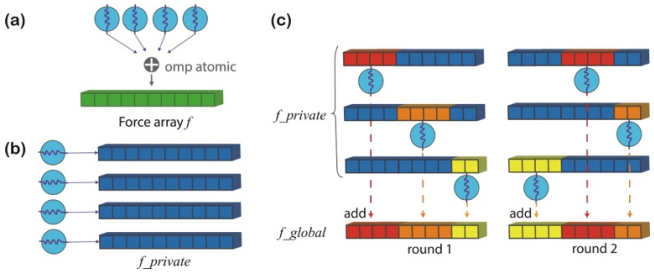

Fig. 5. (a, b) Data privatization. All threads write to the global force array using atomic operations (a), or each thread writes to private array (b). (c) Round-robin reduction with non-overlapping chunks for improving data privatization.

Once the force module finishes all of its computation, we need to aggregate the partial results to the global array, for which we adopt a round-robin approach. As shown in Fig. 5(c), *f_private* is divided into p chunks of equal length except for the last one for a remainder, where p is the number of threads. In every round, each thread is in charge of a unique chunk and transfers data to the global array at the corresponding location. It then circles through the array chunk by chunk until the entire *f_private* is transferred. This data reduction is implemented in a single-instruction multiple-data (SIMD) manner. The non-overlapping chunk design prevents memory conflicts, while maintaining thread scalability by enhancing cache locality. Though data privatization requires additional memory allocation that grows as $O(np)$ (p is the number of threads), it is justified for the target application of computational synthesis. Here, minimizing T2S is essential for describing long-time growth process, rather than simulating the largest possible system to saturate the memory.

C. Promoting Data Parallelism by Data Alignment

The Coulombic energy is updated in every PQEq+ iteration. In the algorithm in Table II, the procedure contains a doubly-nested loop over atoms i and j. The outer loop over i is mapped to parallel threads via OpenMP loop parallelism. The inner loop over j in the original RXMD code was executed sequentially. To promote data parallelism (SIMD) on AVX-512 vector lanes (see the experimental platform in section IV.A), we apply SIMD operations on j loop. However, atoms j in the neighbor list of atom i are irregularly stored, and accordingly successive data accesses by index j are non-contiguous in memory, resulting in inefficient cache performance. Here, we notice two facts: PQEQ+ neighbor list is only updated once every time step (the outmost loop); and the index in neighbor list is closely distributed, albeit randomly stored, in a number of groups. To take advantage of this property, we here use quicksort to rearrange the neighbor list at every PQEq+ initialization phase such that atoms j are sorted in ascending order, thus irregular access is avoided. Fig. 6 illustrates the modified neighbor list.

After sorting, groups of atoms with sequential indices are adjacently placed in memory, thus the subsequent computation can leverage SIMD operations on our computing platform.

TABLE II. PQEq get_hsh procedure in RXMD.

```
1.  do atoms i
2.    do atoms j in i's PQEQ+ neighbor
3.      update Coulombic energy
```

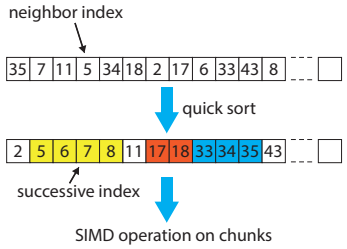

Fig. 6. Sorted neighbor-list followed by SIMD operation.

Table III shows a piece of sorted neighbor list taken from an actual simulation and its corresponding stride. The unit stride (value of 1) indicates successive placement of indices.

TABLE III. Sorted polarized neighbor list and stride.

Index in Neighbor-list
[888, 889, 890, 891, 892, 893, 894, 895, 896, 954, 962, 978, 998, 1002, 1017, 1018, 1024, 1096, 1112, 1145, 1146, 1160, 1180, 1192, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1228, 1232, 1234, 1235, 1236, 1237, 1238, 1239, 1240, 1241, 1242, 1243, 1244]
Stride
[1, 1, 1, 1, 1, 58, 8, 16, 20, 4, 15, 1, 6, 72, 16, 33, 1, 14, 20, 12, 26, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

D. Additional Code Transformations and Performance Improvement

In the original RXMD code, the distance between atom pairs are computed when creating a neighbor list, but this data is discarded immediately regardless the fact that such distance will be frequently recalculated in the subsequent force module. To eliminate these redundant computations, we modify the data structure by packing the atomic-pair distance into the neighbor list. Table IV illustrates the modification of neighbor list and the later usage of the distance data in the force-computation module. When accessing the index of neighboring atoms, the program can directly fetch $d\mathbf{r}_{ij}$ from memory without any redundant computation impairing the performance.

TABLE IV. Packing atomic-pair distance into neighbor list.

```
In Neighbor-list module
1.  for each cell
2.    for atoms i in this cell
3.      for atom j in the supercell
4.        compute distance dr_ij of i and j
5.        if dr_ij < cutoff range
6.          nbrlist(i) stores (j, i, dr_ij)
In Force module, 2-body case
9.  for atom i
10.   for atom j in nbrlist(i)
11.     fetch dr_ij from nbrlist
12.     force calculation using dr_ij
```

To assess the effect of several performance optimizations discussed in this section, we run a simulation of MoS_2 crystal on 512 Intel Xeon Phi KNL nodes with 10 simulation steps. Fig. 7 compares the wall-clock time of the original code (left) with that of our optimized version (right) over a set of ReaxPQ+ force functions, including Enbond (nonbonded), Elnpr (lone-pair), Ehb (hydrogen-bond), E3b (3-body) and E4b (4-body) interactions. The most time-consuming function, E3b, has become 5.57 times faster, and the aggregate time of force-computation module has achieved 4.38-fold speedup over the original version. Overall, these performance optimizations have achieved 1.55-fold speedup of the entire simulation runtime.

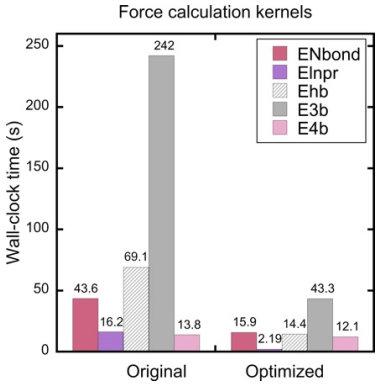

Fig. 7. Performance improvement after data privatization and other code transformations. The numbers denote runtimes.

IV. Scalability and Time-to-Solution

A. Experimental Platform

We perform benchmark tests of the RXMD code with the accelerated ReaxPQ+ model on Theta — an Intel Xeon Phi Knights Landing (KNL) based supercomputer [21]. Each computing node contains 64 cores. Each core in turn has 32 KB L1 instruction cache, 32 KB L1 data cache, and two independent floating-point units capable of executing 512-bit wide SIMD instructions. The peak instruction throughput of the KNL microarchitecture is 2 instructions per clock cycle, and they can be issued back-to-back from the same thread. Two cores and a shared 1 MB L2 cache form a tile, and the tiles are connected with a 2D-mesh network-on-chip with 16 GB of high-bandwidth in-package multichannel DRAM memory (MCDRAM) and 192 GB of regular DRAM. There are two memory modes of execution. In the cache mode, the MCDRAM is used as a cache to DRAM; while in the flat mode, both MCDRAM and DRAM form a single linear memory space. We tested our code in both flat and cache modes but observed no significant difference. Following benchmarks are performed in cache mode.

B. Weak and Strong Scaling on Theta

We perform an isogranular-scaling test of the ReaxPQ+ adapted RXMD code with hybrid MPI+OpenMP implementation on Theta, in which the number of atoms per MPI rank N/P is kept constant. We measure the wall-clock time per simulation time step with scaled workloads — 24,576-atom MoS_2 system on each core. By increasing the number of atoms linearly with the number of cores, the wall-clock time remains almost constant, indicating excellent scalability. To quantify the parallel efficiency, we first define the speed of the RXMD code as a product of the total number of atoms and the number of RMD time steps executed per second. The isogranular (or weak-scaling) speedup is given by the ratio between the speed of P core and that of 64 cores as a reference system. With the granularity of 24,576 atoms per core, the parallel efficiency is 0.989 on 131,072 cores for a 3,221,225,472-atom system, shown in Fig. 8(a). This demonstrates a very high scalability of the ReaxPQ+ adapted RXMD code.

We next perform a strong-scaling test by simulating MoS_2 with a total of 50,331,648 atoms. In this test, the number of cores ranges from P = 2,048 to 32,768, while keeping the total problem size constant. We measure the wall-clock time per RMD time step as a function of P cores. The runtime is reduced by a factor of 12.26 on 32,768 cores compared with the 2,048 cores run (*i.e.*, using 16-times larger number of cores). This signifies a strong-scaling speedup of 12.26, with the corresponding strong-scaling parallel efficiency of 0.766. Fig. 8(b) shows the measured strong-scaling speedup as a function of the number of ranks (blue line), while the black line denotes the ideal speedup. It is more difficult to achieve high strong-scaling parallel efficiency compared with weak-scaling parallel efficiency, as the comparison of Fig. 8, (a) and (b), suggests. This is due to the surge of communication/computation ratio as the workload per rank shrinks proportionally. With 64 times smaller system size of the weak-scaling test, the observed strong-scaling parallel efficiency is considered excellent.

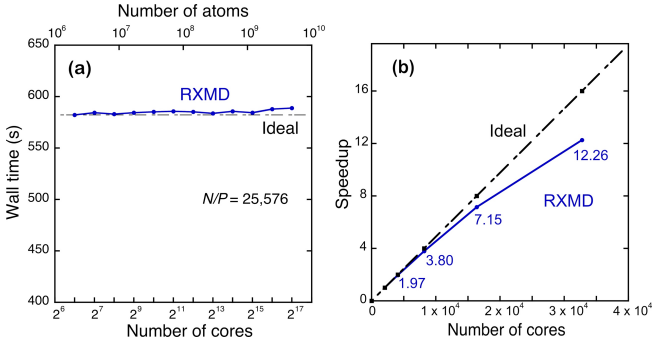

Fig. 8. (a) Wall-clock time of the ReaxPQ+ adapted RXMD code, with scaled workloads — 24,576P-atom MoS_2 on P cores (P = 64, ..., 131,072) of Theta. (b) Strong-scaling speedup of the ReaxPQ+ adapted RXMD code with a fixed problem size — 50,331,648-atom MoS_2 system on P cores (P = 2,048, ..., 32,768) of Theta. The measured speedup values (blue) are compared with the ideal speedup (black). The numbers denote speedups.

C. Time-to-Solution Improvement

Our target application (see section V) is computational synthesis of MoS_2 monolayer from MoO_3 and S_2 reactants. Due to the long time of the reaction process, no previous RMD simulation has been able to complete the reaction to obtain the MoS_2 product. Reduced T2S is the key figure of merit for this purpose. Overall, the algorithmic acceleration in section II has resulted in a speedup of factor 5.0 over the original RXMD code for the ReaxPQ+ model. In addition, a series of code transformations on Theta in section III has achieved additional speedup of factor 1.55. Overall, the algorithmic acceleration and performance optimization have achieved a speedup of factor $5.0 \times 1.55 = 7.75$. Namely, T2S has been reduced to 12.9% compared to that of the original ReaxPQ+ adapted RXMD code.

V. Machine-Learning Guided Computational Synthesis of Atomically-Thin Layered Materials

The accurate description of atomic charges and polarization by the new ReaxPQ+ model, combined with the drastic T2S reduction due to algorithmic acceleration and performance optimization in the previous section, has opened a new avenue for computational synthesis of novel materials. This section demonstrates the capability of the resulting parallel RMD code for computational synthesis of atomically thin layered materials.

We focus on atomically-thin layered materials (LMs) that have unique electronic structures and mechanical and transport properties not found in their three-dimensional counterparts, which makes them attractive templates for future functional devices [22]. The primary synthesis technique for fabrication of LMs is chemical vapor deposition (CVD), where one or more reaction precursors in the gas phase undergo chemical reactions at elevated temperatures inside a reaction chamber and the reaction product is deposited on a substrate in a colder region of the substrate [23]. RMD simulations can provide valuable inputs to rational optimization of CVD growth conditions if adequate length (1,000 Å) and time (10 ns) scales can be covered.

We simulated CVD synthesis of molybdenum disulfide (MoS_2), a prototypical 2D semiconductor, on 242,144 Intel Xeon Phi cores. MoS_2 can be formed by the reaction of MoO_3 and S_2 precursors. The initial configuration (Fig. 9(a)) consists of ~1.14 million atoms (129,472 Mo, 396,032 O and 620,032 S atoms) in a 1,037 Å × 1,080 Å × 145 Å simulation cell. Fig. 9, (b) and (c), shows computational synthesis of MoS_2 monolayer by CVD and subsequent annealing. A pre-sulfurized MoO_3 sample was thermalized at 3,000 K, and then quenched to 1,000 K in 2.2 ns. During annealing, the system was thermalized at 1,500 K for 2 ns, then quenched to 1,000 K in 1 ns. We repeated the annealing cycle twice.

We used a machine-learning approach to identify key reaction pathways. Fig. 9(d) shows the feed-forward neural network (FNN) model [24, 25] we have developed to identify and classify atomic structures into 1T-crystalline (green), 2H-crystalline (red) and disordered (blue) structures in the synthesized MoS_2 crystal. In the input layer, the local environment for each atom is represented by a 60-dimension feature vector consisting of radial and angular symmetry functions [26]. The first, second and third hidden layers consist of 350, 100 and 50 hidden units, respectively. The RELU activation function was used in the first and second layers, while a sigmoid function in the third layer. We trained the model using

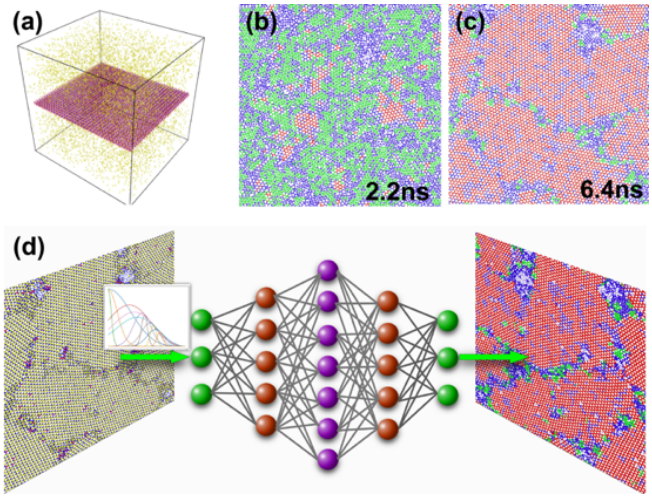

Fig. 9. Machine learning-guided computational synthesis of MoS_2 monolayer by CVD. (a) Simulation setup showing a MoO_3 monolayer suspended in S_2 gas. The atoms are colored as Mo: blue, S: yellow, and O: red. (b, c) Close-up of MoS_2 monolayer before (b) and after (c) annealing. The local structures are classified into 1T (green), 2H (red) and disordered (blue) phases. For clarity, gaseous environment is not shown. (d) Neural network model for defect identification and classification.

36,000-simulation datasets by minimizing the SoftMax function using Adam-optimizer. The results reveal a novel growth mechanism of 2H crystal mediated by a metastable 1T crystalline phase (Fig. 9(b)). Such atomistic information is indispensable for guiding experimental CVD synthesis with improved crystallinity.

VI. Conclusion: Broader Applications on Future Systems

To perform large RMD simulations incorporating dielectric reorganization of materials, we have proposed a new generalized polarizable reactive force-field (ReaxPQ+) model. The increased accuracy of ReaxPQ+, along with the reduced time-to-solution achieved by algorithmic and computational innovations, has for the first time allowed purely computational synthesis of atomically-thin transition-metal dichalcogenide layers assisted by machine learning. This new capability opens up an exciting possibility of future computational synthesis of yet-to-exist layered materials with desired properties. As such, layered materials genome has been chosen as one of the 10 designated applications of the United States' first exaflop/s computer named A21 when it is introduced in 2021 [13]. The computational approaches developed in this paper will likely play an important role in the exascale materials genome. Since ReaxPQ+ is applicable to a wide variety of elements in the periodic table, the current approach applies to much broader applications in science and engineering.

Acknowledgment

This work was supported as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Award Number DE-SC0014607. An award of computer time was provided by the Aurora Early Science Program. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

References

[1] T. P. Senftle, S. Hong, M. M. Islam, S. B. Kylasa, Y. Zheng, Y. K. Shin, *et al.*, "The ReaxFF reactive force-field: development, applications and future directions," *npj Computational Materials,* vol. 2, p. 15011, Mar 4 2016.

[2] A. K. Rappe and W. A. Goddard, "Charge equilibration for molecular-dynamics simulations," *Journal of Physical Chemistry,* vol. 95, pp. 3358-3363, Apr 18 1991.

[3] A. Nakano, "Parallel multilevel preconditioned conjugate-gradient approach to variable-charge molecular dynamics," *Computer Physics Communications,* vol. 104, pp. 59-69, Aug 1997.

[4] K. Nomura, R. K. Kalia, A. Nakano, and P. Vashishta, "A scalable parallel algorithm for large-scale reactive force-field molecular dynamics simulations," *Computer Physics Communications,* vol. 178, pp. 73-87, Jan 15 2008.

[5] H. M. Aktulga, S. A. Pandit, A. C. T. van Duin, and A. Y. Grama, "Reactive molecular dynamics: numerical methods and algorithmic techniques," *SIAM Journal on Scientific Computing,* vol. 34, pp. C1-C23, Jan 31 2012.

[6] S. B. Kylasa, H. M. Aktulga, and A. Y. Grama, "Reactive molecular dynamics on massively parallel heterogeneous architectures," *IEEE Transactions on Parallel and Distributed Systems,* vol. 28, pp. 202-214, Jan 1 2017.

[7] K. Nomura, R. K. Kalia, Y. Li, A. Nakano, P. Rajak, C. Sheng, *et al.*, "Nanocarbon synthesis by high-temperature oxidation of nanoparticles," *Scientific Reports,* vol. 6, p. 24109, Apr 20 2016.

[8] R. A. Marcus, "Electron-transfer reactions in chemistry - theory and experiment," *Reviews of Modern Physics,* vol. 65, pp. 599-610, Jul 1993.

[9] S. Naserifar, D. J. Brooks, W. A. Goddard, and V. Cvicek, "Polarizable charge equilibration model for predicting accurate electrostatic interactions in molecules and solids," *Journal of Chemical Physics,* vol. 146, p. 124117, Mar 28 2017.

[10] K. Nomura, P. E. Small, R. K. Kalia, A. Nakano, and P. Vashishta, "An extended-Lagrangian scheme for charge equilibration in reactive molecular dynamics simulations," *Computer Physics Communications,* vol. 192, pp. 91-96, July 2015.

[11] M. Kunaseth, R. K. Kalia, A. Nakano, K. Nomura, and P. Vashishta, "A scalable parallel algorithm for dynamic range-limited n-tuple computation in many-body molecular dynamics simulation," *Proceedings of Supercomputing, SC13,* ACM/IEEE, 2013.

[12] A. K. Geim and I. V. Grigorieva, "Van der Waals heterostructures," *Nature,* vol. 499, pp. 419-425, Jul 25 2013.

[13] R. F. Service, "Design for US exascale computer takes shape," *Science,* vol. 359, pp. 617-618, Feb 9 2018.

[14] F. Shimojo, R. K. Kalia, M. Kunaseth, A. Nakano, K. Nomura, S. Ohmura, *et al.*, "A divide-conquer-recombine algorithmic paradigm for multiscale materials modeling," *Journal of Chemical Physics,* vol. 140, p. 18A529, May 14 2014.

[15] P. Umari and A. Pasquarello, "Ab initio molecular dynamics in a finite homogeneous electric field," *Physical Review Letters,* vol. 89, p. 157602, Oct 7 2002.

[16] A. M. N. Niklasson, "Extended Born-Oppenheimer molecular dynamics," *Physical Review Letters,* vol. 100, p. 123004, Mar 28 2008.

[17] P. Souvatzis and A. M. N. Niklasson, "First principles molecular dynamics without self-consistent field optimization," *Journal of Chemical Physics,* vol. 140, p. 044117, Jan 28 2014.

[18] D. C. Rapaport, *The Art of Molecular Dynamics Simulation*, Second ed. Cambridge, UK: Cambridge University Press, 2004.

[19] D. E. Shaw, "A fast, scalable method for the parallel evaluation of distance-limited pairwise particle interactions," *Journal of Computational Chemistry,* vol. 26, pp. 1318-1328, Oct 2005.

[20] M. Kunaseth, R. K. Kalia, A. Nakano, P. Vashishta, D. F. Richards, and J. N. Glosli, "Performance characteristics of hardware transactional memory for molecular dynamics application on BlueGene/Q: toward efficient multithreading strategies for large-scale scientific applications," *Proceedings of the International Workshop on Parallel and Distributed Scientific and Engineering Computing, PDSEC-13,* Mar 20 IEEE, 2013.

[21] A. Sodani, R. Gramunt, J. Corbal, H. S. Kim, K. Vinod, S. Chinthamani, *et al.*, "Knights Landing: second-generation Intel Xeon Phi product," *IEEE Micro,* vol. 36, pp. 34-46, Mar-Apr 2016.

[22] K. S. Novoselov, A. Mishchenko, A. Carvalho, and A. H. C. Neto, "2D materials and van der Waals heterostructures," *Science,* vol. 353, p. aac9439, Jul 29 2016.

[23] Y. M. Shi, H. N. Li, and L. J. Li, "Recent advances in controlled synthesis of two-dimensional transition metal dichalcogenides via vapour deposition techniques," *Chemical Society Reviews,* vol. 44, pp. 2744-2756, May 7 2015.

[24] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks,* vol. 2, pp. 359-366, Jan 1989.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research,* vol. 15, pp. 1929-1958, Jun 2014.

[26] E. D. Cubuk, S. S. Schoenholz, J. M. Rieser, B. D. Malone, J. Rottler, D. J. Durian, *et al.*, "Identifying Structural Flow Defects in Disordered Solids Using Machine-Learning Methods," *Physical Review Letters,* vol. 114, p. 108001, Mar 2015.